AMSTRAD

CP/M 2.2

ANATOLE D'HARDANCOURT



AMSTRAD

CP/M 2.2

ANATOLE D'HARDANCOURT

AMSTRAD CP/M 2.2



Paris • Berkeley • Düsseldorf

La Figure 2. 1 est extraite du *Guide du CP/M* de Rodnay Zaks, publié aux éditions Sybex.
Les annexes 1 et 11 sont extraites de *Amstrad CPC 664, manuel de l'utilisateur,* et publiées avec l'aimable autorisation de Amsoft/Amstrad.

CP/M est une marque déposée de Digital Research Inc.

Tous les efforts ont été faits pour fournir dans ce livre une information complète et exacte. Néanmoins, Sybex n'assume de responsabilités, ni pour son utilisation, ni pour les contrefaçons de brevets ou atteintes aux droits de tierces personnes qui pourraient résulter de cete utilisation.

ISBN 2-7361-0156-6

S O M M A I R E

3
LES COMMANDES DE CP/M

4
LA COMMANDE PIP

5

L'ÉDITEUR DE TEXTE ED

6

L'ASSEMBLEUR ASM

7

LE PROGRAMME DE MISE AU POINT DDT

ANNEXES

PRÉFACE

Quand, à l'automne 1984, Amstrad lança sur le marché son premier ordinateur personnel, le CPC 464, personne n'imaginait le succès foudroyant qui allait suivre : plusieurs dizaines de milliers de machines vendues en six mois. Ce succès était dû à trois raisons majeures : son prix, ses caractéristiques techniques et sa disponibilité immédiate. Le CPC 464 offrait en effet, pour moins de 3 000 francs, en version monochrome, un microprocesseur puissant et rapide (Z80 à 4 MHz), 64 K de mémoire vive, un BASIC exceptionnel, un graphisme étonnant (640 × 200), un clavier professionnel, un lecteur/enregistreur de cassettes au fonctionnement irréprochable et un moniteur intégrant l'alimentation du système. De plus, une gamme très diversifiée de logiciels était disponible en même temps que la machine. Hormis le lecteur de cassettes, cette configuration était identique aux matériels professionnels de haut de gamme disponibles seulement deux années auparavant.

Cependant, malgré l'existence de nombreux logiciels professionnels de haut niveau, les acheteurs se cantonnaient plutôt à l'utilisation des nombreux jeux de très haute qualité ou à la programmation en BASIC. L'usage du lecteur de cassettes constituait un obstacle réel à une utilisation professionnelle. Cet obstacle tombait au printemps 1985 avec la sortie de l'Amstrad CPC 664 et l'annonce du CPC 6128, reprenant la majeure partie des caractéristiques qui avaient fait le succès du modèle précédent, mais intégrant cette fois un lecteur/enregistreur de disquettes trois pouces. Cette fois, pour moins de 4 500 francs, les utilisateurs avaient à leur disposition une configuration réellement professionnelle qui, deux ans plus tôt, aurait coûté dix fois plus cher. De plus était livré avec la machine le système d'exploitation CP/M, véritable standard mondial de la micro-informatique professionnelle, permettant à l'Amstrad d'utiliser plusieurs milliers de logiciels. Cependant, si CP/M est un système d'exploitation universel extrêmement puissant, son utilisation par des personnes ne connaissant rien à l'informatique peut se révéler au début assez déconcertante. Elle est en fait très simple une fois les principes de base bien assimilés. Le but de cet ouvrage

est d'offrir au lecteur un guide de référence complet du système CP/M version 2.2 afin de lui permettre de tirer le meilleur parti des possibilités presque infinies de son ordinateur.

Le Chapitre 1 présente le matériel et les différentes précautions à prendre avant la première utilisation.

Le Chapitre 2 constitue une brève introduction à CP/M et à l'utilisation des commandes de base : démarrage, manipulation de fichiers et copie de disquettes.

Le Chapitre 3 est un répertoire complet des commandes CP/M présentées par ordre alphabétique avec leurs conditions d'utilisation et leurs paramètres.

Le Chapitre 4 présente l'utilitaire de manipulation de fichier PIP. Les notions d'unités logiques et physiques sont également étudiées dans ce chapitre.

L'éditeur de texte ED est étudié au Chapitre 5, l'assembleur ASM au Chapitre 6 et le programme de mise au point DDT au Chapitre 7.

Douze annexes regroupent en fin d'ouvrage toutes les informations nécessaires pour une utilisation aisée de CP/M.

1

LE MATÉRIEL

L'Amstrad est livré en deux configurations : avec écran couleur ou avec écran monochrome. L'utilisation des deux versions est identique. La première chose à faire après avoir déballé le matériel est de contrôler son bon fonctionnement et de vérifier la présence de tous les fichiers sur la disquette système livrée avec l'ordinateur.

BRANCHEMENT DU MATÉRIEL

Un des avantages de l'Amstrad est la netteté du système lui donnant un aspect qui confirme sans équivoque le côté professionnel de ses caractéristiques. En effet, fini le temps des ordinateurs " spaghettis". Le moniteur de l'Amstrad intègre l'alimentation nécessaire à l'unité centrale et à l'unité de disquettes. L'installation se résume à la connexion des fiches d'alimentation 12 volts (CPC 664 et CPC 6128) et 5 volts (toutes versions) et à celle de la prise vidéo. Aucune confusion possible, la fiche 12 volts se connectant sur le moniteur et la fiche 5 volts sur l'unité centrale. La prise vidéo se connectant sur l'unité centrale est de type DIN et ne peut être confondue avec aucune des deux autres. Les lecteurs de disquettes externes se connectent également à l'arrière de l'unité centrale. Le CPC 464 peut en recevoir deux : une unité de type DD1, vendue avec un contrôleur, et une unité supplémentaire de type FD1, livrée sans contrôleur (le contrôleur étant capable de gérer deux unités). Le CPC 664 et le CPC 6128 ne peuvent recevoir qu'une seule unité de type FD1 (sans contrôleur), l'unité centrale intégrant un contrôleur capable de gérer une unité supplémentaire. Dans les deux cas, la configuration maximale est de deux unités de disquettes.

Quelques précautions doivent être prises lors de l'installation du matériel. La disposition des différents éléments doit en général répondre aux critères de confort de l'utilisateur : clavier suffisamment bas pour réduire la fatigue des bras lors de la frappe, moniteur aussi éloigné de l'unité centrale que le permet la longueur des câbles de liaison pour la sécurité des yeux (aux éclairages de face ou de dos, préférez un éclairage latéral afin d'éviter les reflets sur l'écran). Une précaution supplémentaire est cependant à observer : les unités de disquettes, comme d'ailleurs les lecteurs de cassettes, doivent être aussi éloignées que possible de toute source de rayonnement électromagnétique. On placera donc l'unité centrale plutôt décalée vers la droite par rapport au moniteur, celui-ci constituant une source de rayonnement intense. On évitera également la proximité des câbles d'alimentation, des lignes électriques, du téléphone, de tout appareil comportant un haut-parleur et, bien sûr,

d'un aimant ou d'un électro-aimant. Les mêmes précautions s'appliquent à l'installation d'une unité de disquettes supplémentaire et à la manipulation des disquettes. En effet, les informations y sont enregistrées sous forme de minuscules champs magnétiques qui risquent d'être perturbés par toute influence magnétique extérieure. Les risques de détérioration des disquettes ne se limitent malheureusement pas aux influences magnétiques. Les disquettes craignent les chocs, la poussière, les liquides, la chaleur, le froid, le contact des doigts ou de tout objet sur leur surface magnétique et bien d'autres choses encore. Deux sortes de précautions peuvent être prises pour la sécurité des disquettes et des informations qu'elles contiennent. (N'oubliez jamais que ce qui fait la valeur d'une disquette, ce sont les informations qu'elle contient. Imaginez la perte qui résulterait de la destruction d'une disquette contenant le résultat d'un mois de travail !) Un premier type de précaution consiste à éviter tout risque de détérioration en respectant les consignes d'utilisation et de stockage (en particulier, remettez toujours les disquettes dans leur boîte en plastique après usage et ne laissez jamais une disquette dans le lecteur lorsque l'ordinateur n'est pas en fonctionnement). Le second type de précaution à prendre *obligatoirement* consiste à tout faire pour réduire au maximum les conséquences d'un accident. Pour cela, une seule solution : faire des copies de sauvegarde. La règle à respecter impérativement est qu'il doit toujours exister une copie de toute disquette manipulée. Les opérations de copie de disquettes nécessitant la manipulation simultanée de deux disquettes (l'original et la copie en cours de réalisation), il doit toujours exister *trois* copies au moins de chaque disquette. Ainsi, si un accident arrive pendant une opération de copie et si l'original et la copie en cours sont détruits, tout n'est pas perdu. Une autre règle veut que toute copie détruite soit *immédiatement* reconstituée. Les copies sont destinées uniquement à la sécurité et ne doivent jamais être utilisées. De plus, si les disquettes contiennent des informations de grande valeur, il faut impérativement ranger les copies dans des endroits différents.

La première opération indispensable va donc être d'effectuer une copie de la disquette système livrée avec l'ordinateur. En effet, cette disquette contient tous les programmes nécessaires à l'utilisation de votre ordinateur (dont CP/M). En cas de destruction de cette disquette, votre ordinateur ne pourrait donc plus fonctionner. Il faudrait alors commander au constructeur de l'appareil une autre disquette système, ce qui risquerait d'immobiliser l'appareil à un moment où vous pourriez en avoir besoin.

Avant de mettre en route l'Amstrad, vérifiez toujours que le lecteur ne contient pas de disquette. Allumer ou éteindre l'ordinateur avec une disquette dans le lecteur risquerait d'entraîner un effacement des informations qui s'y trouvent enregistrées. Vous devrez donc également toujours retirer la disquette du lecteur avant d'éteindre l'ordinateur.

Après avoir vérifié qu'aucune disquette ne se trouve dans le lecteur, actionnez l'interrupteur à poussoir se trouvant en bas à droite de la face avant du moniteur, ce qui a pour effet de mettre celui-ci sous tension. Placez ensuite l'interrupteur se trouvant sur la face latérale droite de l'ordinateur (côté lecteur de disquettes) sur la position ON (sur le CPC 6128, l'interrupteur se trouve sur la face arrière). L'écran affiche :

```
Amstrad 64K Microcomputer (v2)
(c) 1984 Amstrad Consumer Electronics plc
              and Locomotive Software Ltd.

BASIC 1.1

Ready
█
```

Ce message indique que l'ordinateur se trouve sous BASIC et est prêt à travailler (sur le 6128, le message indique 128 K de mémoire et un copyright 1985). Le rectangle de couleur se trouvant sous le message Ready est le curseur. Il indique à quel endroit sera affiché le prochain caractère tapé.

Insérez maintenant la disquette système dans le lecteur, étiquette vers vous et face 1 (CP/M) vers le haut (face 3 sur le 6128). La disquette doit s'enclencher en faisant ressortir le poussoir d'éjection en bas à droite du lecteur. Tapez :

```
¦cpm <ENTER>
```

Le signe ¦ s'obtient en maintenant la touche SHIFT (majuscule) enfoncée et en tapant la touche @/¦, à droite de la touche P. <ENTER> signifie que vous devez presser la touche ENTER. Sur le CPC 6128, la touche ENTER peut être remplacée par la touche RETURN, qui a la même fonction et une position plus pratique.

Le lecteur de disquettes se met en marche. Le fonctionnement du lecteur est indiqué par l'allumage du voyant rouge, sur sa face

avant. Vous ne devez *jamais* toucher à la disquette se trouvant dans le lecteur lorsque celui-ci est en fonctionnement (voyant rouge allumé). L'écran affiche maintenant :

```
CP/M 2.2 - Amstrad Consumer Electronics plc
A>▌
```

Le fond de l'écran est devenu bleu clair et les caractères noirs. Notez que ceux-ci sont plus étroits que précédemment. L'ordinateur passe en effet automatiquement en mode d'affichage 80 colonnes lors de l'appel de CP/M.

La deuxième ligne affichée sur l'écran indique deux choses :

A indique que l'unité de disquettes A est active. C'est l'unité par défaut. C'est à elle que s'appliqueront toutes les commandes ne comportant pas d'indication d'unité de disquettes.

> est l'indicatif de CP/M. Il est suivi du curseur qui indique toujours l'emplacement du prochain caractère tapé.

Vous allez maintenant vérifier que votre disquette système contient bien tous les fichiers nécessaires. Tapez :

```
A>DIR <ENTER>
```

Vous ne devez jamais taper le A et le signe > qui sont affichés par l'ordinateur. <ENTER> indique qu'il faut taper la touche portant l'inscription ENTER et non les touches E, N, T, E et R. La commande DIR, abréviation de DIRectory (qui veut dire répertoire en anglais), affiche la liste des fichiers contenus sur la disquette. Les commandes peuvent être tapées indifféremment en majuscules ou en minuscules, ou dans un mélange des deux. Toute commande est en effet traduite en majuscules avant d'être interprétée par CP/M. L'écran doit afficher :

```
A>DIR
A: MOVCPM    COM : PIP       COM : SUBMIT   COM : XSUB      COM
A: ED        COM : ASM       COM : DDT      COM : LOAD      COM
A: STAT      COM : DUMP      COM : DUMP     ASM : AMSDOS    COM
A: FILECOPY  COM : SYSGEN    COM : BOOTGEN  COM : COPYDISK  COM
A: CHKDISC   COM : DISCCOPY  COM : DISCCHK  COM : SETUP     COM
A: FORMAT    COM : CSAVE     COM : CLOAD    COM : EX1       BAS
A: EX2       BAS : ROINTIME  DEM : RITDEMO  BIN : DISC      BAS
A>▌
```

La disquette du CPC 6128 contient en plus les fichiers LOGO2.COM, LOGO2.SUB, DRLKEYS.COM, FWRESET.COM et DISCKIT2.COM.

Si, au lieu d'afficher la liste des fichiers contenus sur la disquette, l'écran réaffiche votre commande suivie d'un point d'interrogation, vous avez fait une faute de frappe. Recommencez en vérifiant que vous ne commettez pas d'erreur.

La seconde face de la disquette système des CPC 464 et 664 contient DR LOGO, un langage de programmation. Pour vérifier le contenu de la face 2, procédez de la façon suivante. Éjectez la disquette du lecteur en appuyant sur le poussoir d'éjection qui se trouve sur la face avant. Si le voyant rouge est allumé, attendez qu'il s'éteigne avant d'éjecter la disquette. Sortez la disquette du lecteur, retournez-la (face 2 vers le haut) et introduisez-la de nouveau en tenant toujours l'étiquette vers vous. Une fois la disquette enclenchée, tapez :

```
A>DIR <ENTER>
```

L'écran doit afficher :

```
A: LOGO    COM : SETUP    COM : AMSDOS  COM
A>▌
```

Éjectez la disquette et insérez-la de nouveau avec la face 1 vers le haut.

COPIE DE LA DISQUETTE SYSTÈME

Pour effectuer la copie de la disquette système, nous allons utiliser la commande DISCCOPY. Cette commande permet d'effectuer une copie de disquette sur un système à un seul lecteur. La commande COPYDISC permet de gagner du temps en effectuant la copie à l'aide de deux lecteurs de disquettes si l'on dispose d'un lecteur supplémentaire. Cette commande sera décrite au Chapitre 3.

La disquette système se trouvant dans le lecteur, face 1 vers le haut (face 3 sur le 6128), et l'écran affichant l'indicatif A>, tapez :

```
A>DISCCOPY <ENTER>
```

L'écran affiche :

```
DISCCOPY V2.0

Please insert source disc into drive A then press any key:_
```

Ce message vous demande de placer la disquette à copier (que l'on appelle *disquette source*) dans le lecteur A puis de taper une touche quelconque. (Sur le CPC 664 et le CPC 6128, le lecteur A est le lecteur intégré à l'ordinateur. Sur le CPC 464, le lecteur A est celui qui est le plus éloigné de l'ordinateur.) Si vous n'obtenez pas ce message, deux causes sont possibles. Vous avez fait une erreur de frappe (recommencez) ou vous avez oublié de retourner la disquette face 1 vers le haut (retournez-la et recommencez).

La disquette à copier se trouvant déjà dans le lecteur A, appuyez simplement sur une touche quelconque (la barre d'espacement par exemple). Le lecteur de disquettes se met en marche pendant que l'écran affiche divers messages puis :

```
Copying started
Please insert destination disc into drive A then press any key:_
```

L'ordinateur indique que la copie a commencé et vous demande d'introduire la disquette qui contiendra la copie (appelée *disquette destination*) dans le lecteur. Retirez la disquette système du lecteur et placez-y une disquette vierge. Une fois cette opération effectuée, tapez une touche quelconque : le lecteur se remet en marche et copie les huit premières pistes de la disquette avant de demander à nouveau l'introduction de la disquette source. Cette opération est répétée cinq fois pour copier entièrement la disquette, puis l'ordinateur affiche le message :

```
Copying complete
Do you want to copy another disc (Y/N):
```

La copie de la première face est terminée. Pour copier la seconde face, répondez Y à cette question. Lorsque l'ordinateur demande de placer la disquette source ou la disquette destination dans le lecteur, insérez-les avec la face 2 vers le haut. Lorsque la copie de la seconde face est terminée, répondez N à la question précédente. L'écran affiche :

```
Please insert a CP/M system disc into drive A then press any key:
```

Tapez une touche quelconque pour retourner sous CP/M. Si la copie ne s'est pas effectuée correctement, l'écran aura affiché le message suivant :

```
WARNING: Failed to copy disc correctly
The destination disc should not be used until it is successfully copied on to
Please insert a CP/M system disc into drive A then press any key:_
```

Ce message vous informe que la copie a échoué et que la disquette destination ne doit pas être utilisée. Placez la disquette système dans le lecteur et tapez une touche quelconque. Deux raisons principales peuvent entraîner un échec de la procédure de copie : la disquette utilisée est détériorée, ou elle est "protégée en écriture". Si la disquette est détériorée, il faut tout simplement en utiliser une autre. Si elle est protégée en écriture, l'écran aura affiché préalablement le message :

```
The destination disc in drive A must be write-enabled
Please insert destination disc into drive A then press any key:_
```

vous laissant l'opportunité de sortir la disquette du lecteur et d'ôter la protection en écriture comme indiqué sur la Figure 1.1.

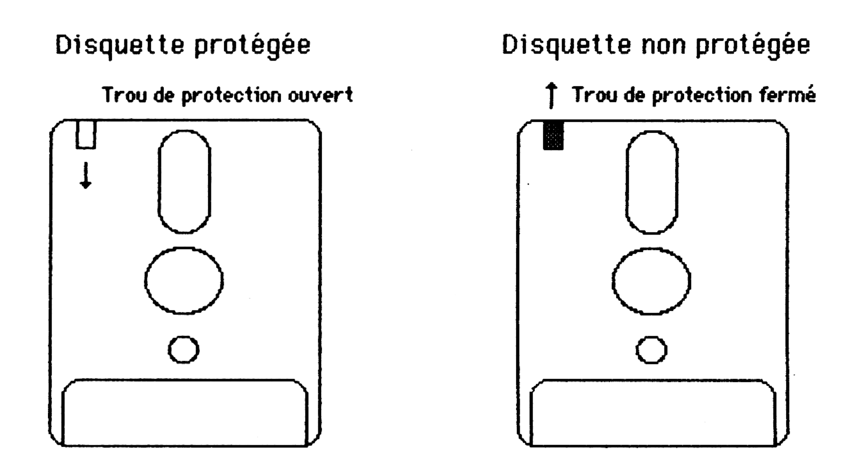


Figure 1.1 : Protection et déprotection d'une disquette.

Une fois les deux faces copiées, nous allons vérifier que la copie a été effectuée correctement. Nous utiliserons pour cela la com-

mande DISCCHK. Placez la disquette système dans le lecteur, face 1 vers le haut, et tapez :

```
A>DISCCHK <ENTER>
```

L'écran affiche le message suivant :

```
DISCCHK V2.0

Please insert source disc into drive A then press any key
```

Placez la disquette originale (disquette source) dans le lecteur A et tapez une touche quelconque. L'ordinateur lit les huit premières pistes de la disquette et affiche un nouveau message :

```
Please insert destination disc into drive A then press any key
```

Insérez la copie (disquette destination) dans le lecteur A et tapez une touche quelconque. La même opération est répétée cinq fois pour comparer les quarante pistes des deux disquettes. Si les deux disquettes sont identiques, la comparaison se termine avec le message :

```
Copy checking complete
Do you want to check another disc (Y/N):
```

Répondez Y et procédez ensuite de la même façon pour vérifier la conformité de la face 2.

Une fois la vérification terminée, placez sur la nouvelle copie une étiquette indiquant son contenu. Rangez soigneusement l'original dans un endroit sûr. Vous ne devrez y recourir qu'en dernière extrémité. Il est donc prudent de réaliser immédiatement une seconde copie de la disquette système (à partir de la première copie) afin d'éviter d'avoir à recourir à l'original en cas de problème.

L'APPEL DE CP/M ET LE RETOUR À AMSDOS

Comme nous l'avons vu précédemment, le passage à CP/M se fait en tapant :

```
|CPM <ENTER>
```

En fait, l'Amstrad est livré avec deux systèmes d'exploitation différents. AMSDOS est un système construit autour de l'excellent BASIC de Locomotive Software et permet à celui-ci d'exploiter les unités de disquettes. CP/M est le standard des systèmes d'exploitation pour ordinateurs personnels 8 bits et permet d'utiliser les centaines de logiciels existant pour ce système. Comme nous le verrons au Chapitre 2, une des caractéristiques de CP/M est son extensibilité. Chaque programme écrit sous CP/M devient une *commande non résidente* de CP/M. Les commandes non résidentes sont celles qui apparaissent sur l'écran avec l'extension .COM lorsque l'on utilise la commande DIR. Comme nous l'avons vu plus haut, il existe sur la disquette système une commande nommée AMSDOS.COM. Cette commande permet de passer de CP/M à AMSDOS. Pour l'utiliser, tapez :

```
A>AMSDOS <ENTER>
```

L'écran affiche alors :

```
Amstrad 64K Microcomputer (v2)
(c)1984 Amstrad Consumer Electronics plc
            and Locomotive Software Ltd.

BASIC 1.1

Ready
█
```

et l'ordinateur est prêt à accepter une commande BASIC.

2

INTRODUCTION A CP/M

Pour tirer le meilleur parti de l'utilisation de CP/M, il faut comprendre les principes de son fonctionnement. CP/M est composé de plusieurs modules distincts.

- La zone des paramètres se trouvant en bas de la mémoire (adresses 0000H à 0100H) contient diverses informations nécessaires au fonctionnement du système (le H après une adresse indique que celle-ci est exprimée en hexadécimal).
- La TPA (*Transient Program Area :* zone des programmes non résidents) est la partie utilisée par tous les programmes tournant sous CP/M.
- Le CCP (*Console Command Processor :* processeur des commandes console) est la zone qui contient les programmes correspondant aux commandes résidentes de CP/M : DIR, ERA, REN, SAVE, TYPE et USER.
- Le BDOS (*Basic Disk Operating System :* système de base de gestion de disques) contient les programmes servant à la gestion des unités logiques (lecteurs de disquettes, imprimante, etc.). Cette partie de CP/M, comme les précédentes, est commune à toutes les machines utilisant ce système.
- Le BIOS (*Basic Input/Output System)* est la partie propre à chaque ordinateur qui permet d'adapter CP/M à des machines différentes en fonction de leurs caractéristiques.

LE DÉMARRAGE A FROID

A l'appel de CP/M (utilisation de la commande ¦CPM de l'AMSDOS), les éléments cités précédemment sont chargés en mémoire. Cette opération est appelée "démarrage à froid". Après chargement de CP/M, le contrôle est passé au BIOS qui remplit les adresses 0000H à 0007H de la zone de paramètres. Le message :

```
A>
```

est affiché sur l'écran et CP/M est prêt à accepter une commande.

CP/M peut normalement gérer seize unités de disquettes. Cependant, l'Amstrad ne peut utiliser que deux unités de disquettes : soit une unité intégrée et une unité externe (cas des CPC 664 et CPC 6128), soit deux unités externes (cas du CPC 464). Les unités de dis-

quettes sont désignées par des lettres (A et B dans un système à deux unités). La lettre A affichée sur l'écran indique que l'unité A est l'unité active ou unité par défaut. Cela signifie que toute commande impliquant l'usage d'une unité de disquettes et ne comportant pas d'indication d'unité utilisera l'unité A. Ainsi, sur un système à deux disquettes, si le message A> est affiché sur l'écran, la commande :

```
A>DIR
```

sera équivalente de :

```
A>DIR A:
```

et affichera sur l'écran la liste des fichiers contenus sur la disquette se trouvant dans l'unité A. La commande :

```
A>DIR B:
```

affichera sur l'écran la liste des fichiers contenus sur la disquette se trouvant dans le lecteur B.

L'indication d'une unité de disquettes dans une commande doit être suivie de deux-points (:) afin d'être différenciée d'un nom de fichier.

Changer le nom de l'unité de disquettes par défaut se fait très simplement en tapant, après l'indicatif de CP/M, le nom de l'unité suivi de deux-points (:) et de la touche ENTER. Ainsi, si l'unité par défaut est A, il suffit, pour passer sur l'unité B, de taper :

```
A>B: <ENTER>
```

L'écran affiche alors :

```
B>
```

et toute commande ne comportant pas d'indication d'unité concernera désormais l'unité B. Ainsi,

```
B>DIR
```

sera équivalent de :

```
B>DIR B:
```

et affichera sur l'écran la liste des fichiers contenus sur la disquette se trouvant dans l'unité B. Pour repasser sur l'unité A, tapez :

B>A: <ENTER>

LES COMMANDES RÉSIDENTES

Six commandes de CP/M sont dites *résidentes* car elles sont partie intégrante du CCP. Il s'agit des commandes suivantes :

Commande	Fonction
DIR	Liste des fichiers se trouvant sur la disquette.
ERA	Effacement d'un fichier.
REN	Changement du nom d'un fichier.
SAVE	Copie d'une zone mémoire sur un fichier.
TYPE	Affichage d'un fichier texte sur l'écran.
USER	Changement du numéro de zone utilisateur.

Ces commandes, à l'inverse des commandes non résidentes qui seront étudiées dans la suite de ce chapitre, ne doivent jamais être précédées de l'indication d'une unité de disquettes. Certaines, par contre, comme la commande DIR déjà citée, peuvent être suivies de l'indication d'une unité si celle-ci est différente de l'unité par défaut. Toutes ces commandes seront étudiées au Chapitre 3.

LES COMMANDES NON RÉSIDENTES

Comme nous l'avons dit, une des particularités de CP/M est d'être extensible à l'infini. Tous les programmes tournant sous CP/M sont en effet considérés comme des commandes d'un type particulier : les commandes non résidentes. Ces commandes figurent sur disquette avec l'extension .COM. Elles sont appelées comme les commandes résidentes par leur nom, éventuellement précédé de l'indication d'une unité de disquettes si elles se trouvent dans une autre unité que celle par défaut. Lorsqu'un nom est tapé après l'indicatif de CP/M, l'ordinateur cherche s'il s'agit d'une commande résidente. Si ce n'est pas le cas, CP/M cherche sur l'unité par défaut, ou sur une autre unité si la commande en comporte l'indication, un programme exécutable correspondant au nom tapé. Si un tel programme existe, son exécution est lancée. Dans le cas contraire,

CP/M réaffiche la commande tapée suivie d'un point d'interrogation indiquant qu'aucun programme de ce nom n'a été trouvé. L'avantage principal des commandes non résidentes est que leur nombre peut être augmenté et leurs possibilités développées sans augmenter la taille de CP/M. En contrepartie, elles doivent, pour être disponibles, se trouver sur une disquette dans l'une des unités. Par contre, les commandes résidentes, moins nombreuses et aux possibilités restreintes, sont accessibles à tout moment, car elles se trouvent en permanence en mémoire centrale.

LE DÉMARRAGE A CHAUD

Un démarrage à chaud est effectué chaque fois qu'un ↑C est tapé et que le curseur se trouve en début de ligne. (Pour taper un ↑C, maintenez la touche CTRL enfoncée et tapez la touche C. Dans la suite de ce livre, et conformément à l'usage de CP/M, les caractères de contrôle seront notés précédés d'une flèche verticale. Par exemple, contrôle-C sera noté ↑C.) Comme lors d'un démarrage à froid, le CCP et le BDOS sont alors copiés en mémoire à partir de la disquette se trouvant dans l'unité A. La zone de paramètres est également réinitialisée, mais le BIOS n'est pas modifié. Lorsque CP/M accède à une disquette pour la première fois (et notamment lors d'un démarrage à froid ou à chaud), une lecture du répertoire de cette disquette est effectuée. (Le répertoire d'une disquette est la zone comportant la liste des fichiers se trouvant sur cette disquette ainsi que toutes les informations permettant d'accéder à ces fichiers.) Cela peut être observé sur un système à deux disquettes. Après un démarrage à froid (passage à CP/M à partir de AMSDOS par exemple), introduisez une disquette contenant des fichiers dans l'unité B. Passez sur l'unité B en tapant :

```
A>B: <ENTER>
```

L'unité de disquettes B se met en marche et le témoin de fonctionnement s'allume. Le répertoire de la disquette se trouvant dans cette unité est chargé en mémoire. L'écran affiche maintenant :

```
B>
```

Revenez sur l'unité A en tapant :

```
B>A: <ENTER>
```

L'écran affiche maintenant :

```
A>
```

Cependant, le témoin de fonctionnement de l'unité de disquettes A ne s'est pas allumé car la disquette se trouvant dans cette unité était déjà connue de CP/M (son répertoire avait été lu lors du démarrage à froid). A toutes les modifications effectuées sur une disquette doivent correspondre des modifications du répertoire. C'est à partir des données lues en mémoire que le répertoire est mis à jour sur la disquette. On imagine aisément ce qui se produirait si l'on changeait de disquette sans en avertir CP/M. Le répertoire de la disquette ne correspondrait plus aux données qui s'y trouvent, rendant toute utilisation impossible. Les informations contenues dans le répertoire sont en effet indispensables pour accéder aux fichiers (les raisons en seront expliquées à la fin de ce chapitre lorsque seront exposées les différences entre copie logique et copie physique de disquettes). Pour éviter ces désagréments, CP/M vérifie avant d'écrire sur une disquette qu'il s'agit bien de celle dont le répertoire a été lu précédemment. Si ce n'est pas le cas, CP/M refuse d'écrire sur la disquette et affiche le message :

```
BDOS ERROR ON A:R/O
```

indiquant que la disquette se trouvant dans l'unité (ici A) ne peut être utilisée qu'en lecture. Cependant, il en résulte un inconvénient qui peut être assez ennuyeux. Supposez que, sur un système à une seule disquette, après un démarrage à froid, vous changiez de disquette pour utiliser un programme de traitement de texte se trouvant sur une autre disquette. Au moment de sauvegarder votre texte, CP/M, ne reconnaissant pas la disquette se trouvant dans l'unité, refusera d'écrire et votre travail sera perdu. Aussi la règle suivante doit-elle impérativement être respectée : effectuez toujours un démarrage à chaud (↑C) après un changement de disquette. Pour effectuer un ↑C, maintenez la touche CTRL enfoncée et pressez la touche C.

LES CARACTÈRES DE CONTRÔLE

Outre le caractère ↑C vu précédemment, CP/M reconnaît plusieurs autres caractères de contrôle.

Caractère	Fonction
↑C	Démarrage à chaud.
↑E	Passage à la ligne suivante.
↑H	Déplacement du curseur vers la gauche avec effacement d'un caractère (équivalent de la touche DEL).
↑I	Tabulation (équivalent de la touche TAB).
↑J	Saut à la ligne. Entraîne l'exécution de la ligne (équivalent de la touche ENTER).
↑M	Retour chariot. Entraîne l'exécution de la ligne (équivalent de la touche ENTER).
↑P	Début ou fin de l'écho sur l'imprimante.
↑R	Réimpression de la ligne en cours.
↑S	Arrêt temporaire de l'affichage.
↑U	Abandon de la ligne en cours. Début d'une nouvelle ligne.
↑X	Abandon de la ligne en cours sans commencer une nouvelle ligne.

↑C est le caractère de contrôle le plus important. Sa principale utilisation a été décrite précédemment. Un autre usage de ↑C est l'interruption d'une commande en cours.

↑E permet de passer à la ligne sans exécuter la commande en cours lorsque la longueur de celle-ci dépasse la largeur de l'écran. Ce caractère est inutile sur l'Amstrad, car le curseur passe automatiquement à la ligne suivante en cas de dépassement de la largeur de l'écran.

↑H est équivalent de la touche DEL et permet de corriger une erreur de frappe en effaçant le caractère situé immédiatement à gauche du curseur.

↑I sert à exécuter une tabulation ; il est équivalent de la touche TAB. Les tabulations sont placées tous les huit caractères.

↑J et ↑M ont sous CP/M le même effet, qui est d'exécuter la ligne de commande en cours. Ces deux caractères sont équivalents de la touche ENTER (et de la touche RETURN sur le CPC 6128).

↑P sert à lancer ou à stopper l'écho sur l'imprimante. Après la frappe d'un premier ↑P, et si l'imprimante est correctement connectée et prête à fonctionner, tout ce qui est affiché sur l'écran est en même temps transmis à l'imprimante. L'écho est supprimé par la frappe d'un second ↑P.

↑S permet de stopper momentanément l'affichage sur l'écran. La frappe d'une touche quelconque (y compris ↑S) relance l'affichage. ↑S sera utilisé pour lire les informations affichées lorsque celles-ci dépassent la hauteur d'un écran.

↑R, ↑U et ↑X sont trois caractères de contrôle utilisés pour faciliter la frappe d'une ligne de commande.

↑R permet de réafficher la ligne en cours. Cette commande est inutile sur l'Amstrad du fait de la possibilité de corriger une faute de frappe en utilisant la touche DEL.

↑U annule la ligne en cours (un # est affiché en fin de ligne) et commence une nouvelle ligne.

↑X annule la ligne en cours et renvoie le curseur en début de ligne sans en commencer une nouvelle.

LES NOMS DE FICHIERS

CP/M manipule les informations sur disquettes sous forme de fichiers. Ces fichiers sont répertoriés dans une zone réservée de chaque disquette appelée répertoire. Le répertoire d'une disquette contient de très nombreuses informations nécessaires pour pouvoir retrouver les fichiers. L'information principale est le nom du fichier. Les noms de fichiers CP/M obéissent à des règles très strictes. Un nom de fichier comprend le nom proprement dit, qui peut avoir jusqu'à huit caractères, et une éventuelle extension de un à trois caractères, séparée du nom par un point et servant généralement à indiquer le type du fichier. Les noms de fichiers peuvent être composés de lettres, de chiffres et de caractères spéciaux. Les caractères suivants sont cependant interdits :

< > . , : = ; * ? [ ]

Les fichiers doivent généralement être désignés par leur nom complet (nom proprement dit plus extension). Une exception est l'utilisation des commandes non résidentes. Pour utiliser une telle

commande, l'extension .COM ne doit pas être tapée. Un certain nombre d'extensions ont une signification particulière :

Extension	Signification
ASM	Programme source en langage assembleur.
BAK	Copie de sécurité d'un fichier portant le même nom, créée par ED ou certains autres programmes.
BAS	Programme BASIC.
BIN	Fichier contenant un programme ou l'image d'une zone de mémoire sauvegardés sous AMSDOS.
COM	Commande non résidente (tout programme exécutable).
HEX	Fichier hexadécimal produit par l'assemblage d'un programme et destiné à être utilisé avec la commande LOAD.
PRN	Fichier d'impression produit par l'assemblage d'un programme.
SUB	Fichier texte contenant une liste de commandes destinées à être exécutées par la commande SUBMIT.
$$$	Fichier temporaire créé par certains programmes (ED par exemple). Ces fichiers sont normalement détruits par le programme qui les crée. La présence d'un fichier $$$ dans un répertoire témoigne d'une interruption anormale d'un programme.

Un nom de fichier peut être complété par l'indication de l'unité de disquettes dans laquelle il se trouve, si celle-ci est différente de l'unité par défaut. La spécification complète d'un fichier est donc de la forme :

```
UNITE : NOM.EXTENSION
```

soit par exemple :

```
B : STAT.COM
```

LES CARACTÈRES GÉNÉRIQUES

Il est souvent utile de pouvoir désigner plusieurs fichiers à la fois. On utilise pour cela les caractères génériques. Ces caractères sont le point d'interrogation (?) et l'astérisque (*).

Le point d'interrogation remplace n'importe quel caractère. Ainsi, si l'on a sur une disquette les fichiers suivants :

```
A:STAT     .COM : LOAD    .COM : ED      .COM : ASM     .COM
A:PIP      .COM : LETTRE1 .TXT : LETTRE1 .BAK : LETTRE2 .TXT
A:LETTRE2  .BAK :
```

la commande

```
A>DIR LETTRE?.TXT
```

affichera :

```
A: LETTRE1 .TXT : LETTRE2 .TXT
```

La commande

```
A>DIR LETTRE?.???
```

affichera :

```
A: LETTRE1 .TXT : LETTRE1 .BAK : LETTRE2 .TXT : LETTRE2 .BAK
```

L'astérisque sert à remplacer un groupe de caractères. Ainsi, la dernière commande utilisée est équivalente de :

```
A>DIR LETTRE?.*
```

Dans le cas présent, cette commande aurait le même effet que :

```
A>DIR LE*.*
```

Par contre

```
A>DIR L*.*
```

produirait l'affichage suivant :

```
A: LOAD    .COM: LETTRE1 .TXT: LETTRE1 .BAK: LETTRE2 .TXT
A: LETTRE2 .BAK:
```

Le fichier LOAD.COM se trouve désigné par L*.* qui signifie en fait "Tous les fichiers dont le nom commence par L et qui ont une extension quelconque ou pas d'extension du tout".

Il faut toujours être très prudent en utilisant les caractères génériques, car désigner tous les fichiers contenus sur une disquette peut être dangereux, en particulier avec la commande ERA. Ainsi :

```
A>ERA *.*
```

efface tous les fichiers contenus sur la disquette se trouvant dans l'unité A, ce qui peut avoir des conséquences fâcheuses en cas d'erreur. CP/M affichera cependant une demande de confirmation :

```
ALL?(Y/N)
```

Le contenu de la disquette ne sera effacé que si l'utilisateur tape Y suivi de la touche ENTER.

CRÉATION D'UN FICHIER

Il existe plusieurs façons de créer un fichier. Presque tous les programmes d'application créent des fichiers. Nous allons utiliser l'éditeur de texte ED pour créer un petit fichier texte.

Introduisez une copie de la disquette système non protégée dans l'unité de disque A et tapez ↑C. Souvenez-vous que vous devez toujours taper ↑C après avoir introduit dans une unité une disquette sur laquelle vous désirez pouvoir écrire. Tapez :

```
A>ED FICHIER.TXT <ENTER>
```

(L'indicatif A> n'est pas à taper mais est affiché par l'ordinateur. <ENTER> représente la frappe de la touche ENTER.)

Le programme ED est chargé en mémoire et, après quelques secondes, l'écran affiche :

```
NEW FILE

*:
```

Tapez :

```
I <ENTER>
```

pour passer en mode insertion de texte. L'écran affiche alors :

```
1*:
```

Vous allez maintenant taper un texte de quelques lignes. Tapez la touche ENTER chaque fois que vous voudrez aller à la ligne. Pour corriger une faute de frappe, utilisez la touche ↑H. Vérifiez bien chaque ligne avant de passer à la suivante, car nous n'étudierons la manière de modifier une ligne déjà tapée qu'au Chapitre 5.

Tapez par exemple le petit texte suivant, extrait de la notice de l'Amstrad :

```
 1 : L'informatique a parcouru un grand chemin en un temps très
 2 : court. De toutes les inventions du 20e siècle, l'ordinateur
 3 : est certainement la plus extraordinaire.
 4 :
 5 : Les capacités des machines et des logiciels ont progressé si
 6 : rapidement que même les utilisateurs habituels ont du mal à
 7 : suivre, et pour montrer aux possesseurs d'un Amstrad tout le
 8 : potentiel et la subtilité de son BASIC, de son système
 9 : d'exploitation et de son architecture il faudrait des milliers
10 : de pages.
11 :
```

Après la dernière ligne de texte, tapez <ENTER> puis ↑Z, E et à nouveau <ENTER>. La disquette se met à tourner pendant quelques secondes et l'indicatif de CP/M apparaît sur l'écran. Vérifiez qu'un fichier de nom FICHIER.TXT a bien été créé sur la disquette en utilisant la commande DIR. Vous constatez en fait que deux fichiers ont été créés. Un fichier nommé FICHIER.TXT, qui est votre fichier texte, et un fichier nommé FICHIER.BAK, qui est la copie de sécurité de votre fichier. L'utilisation des fichiers .BAK sera détaillée au Chapitre 5.

Nous allons vérifier le contenu de ces deux fichiers. Nous utilise-

rons pour cela la commande résidente TYPE suivie du nom du fichier à visualiser. Cette commande a pour effet d'envoyer sur l'écran le contenu d'un fichier. Elle ne doit être employée qu'avec des fichiers dont le contenu est affichable, c'est-à-dire des fichiers texte, des listings d'assemblage ou des fichiers hexadécimaux par exemple. Utiliser cette commande avec des fichiers .COM peut produire des résultats imprévisibles.

Tapez :

```
A>TYPE FICHIER.TXT <ENTER>
```

(N'oubliez pas l'extension.) Le contenu du fichier s'affiche sur l'écran. Vous pouvez arrêter temporairement l'affichage en tapant ↑S ou définitivement en tapant une touche quelconque. Pour relancer un affichage interrompu par ↑S, tapez une touche quelconque. Tapez maintenant :

```
A>TYPE FICHIER.BAK <ENTER>
```

Le texte affiché sur l'écran est identique au texte précédent.

CHANGEMENT DU NOM D'UN FICHIER

Nous allons maintenant changer le nom de ce fichier en utilisant la commande résidente REN. La syntaxe de cette commande est de la forme :

REN <nouveau nom> = <ancien nom>

Tapez :

```
A>REN TEXTE.TXT=FICHIER.TXT <ENTER>
```

La disquette se met à tourner pendant quelques secondes et l'indicatif de CP/M réapparaît. Vérifiez que le nom du fichier FICHIER.TXT a bien été changé en TEXTE.TXT en utilisant la commande DIR.

On peut utiliser des caractères génériques avec la commande REN. Commencez par rétablir le nom original de votre fichier en tapant :

```
A>REN FICHIER.TXT=TEXTE.TXT
```

(Nous n'indiquerons plus désormais la touche <ENTER>, qui doit être tapée pour lancer toute commande.)

Nous avons donc maintenant à nouveau les deux fichiers FICHIER.TXT et FICHIER.BAK. Tapez :

```
A>REN TEXTE.*=FICHIER.*
```

Vérifiez à l'aide de la commande DIR que les noms des deux fichiers ont bien été changés en TEXTE.TXT et TEXTE.BAK. Vous pouvez également utiliser la commande TYPE pour vérifier que leur contenu n'a pas été modifié.

COPIE D'UN FICHIER

Nous allons maintenant effectuer une copie du fichier TEXTE.TXT. Nous utiliserons pour cela la commande non résidente PIP dont la syntaxe est :

```
PIP <nom de la copie>=<nom du fichier à copier>
```

Tapez :

```
A>PIP LETTRE.TXT=TEXTE.TXT
```

(N'oubliez pas de taper la touche ENTER.) .

Vérifiez ensuite à l'aide de la commande DIR que le fichier LETTRE.TXT a bien été créé sur la disquette. Affichez son contenu sur l'écran à l'aide de la commande TYPE.

La commande PIP permet également, sur un système à deux unités de disquettes, de copier un fichier d'une disquette sur une autre. (La commande FILECOPY, qui permet d'arriver au même résultat sur un système à une seule disquette, est décrite au Chapitre 3.) Pour cela, nous aurons besoin d'une seconde disquette. Les disquettes vendues dans le commerce ne sont pas prêtes à l'emploi. Elles doivent être formatées à l'aide de la commande FORMAT (cette commande est décrite en détail au Chapitre 3). Placez une disquette formatée dans l'unité de disquettes B, la disquette système se trouvant dans l'unité A. Tapez :

```
A>PIP B:=A:LETTRE.TXT
```

On reconnaît la syntaxe :

```
PIP <nom de la copie> = <nom du fichier à copier>
```

Cependant, seule l'unité de disquettes a été mentionnée dans le nom de la copie. De cette façon, la copie portera le même nom que l'original. Cette commande est en fait équivalente de la commande :

```
PIP B:LETTRE.TXT=A:LETTRE.TXT
```

On vérifiera facilement, en affichant le contenu du fichier à copier, que l'opération de copie ne modifie en rien l'original.

IMPRESSION D'UN FICHIER

La commande PIP permet également de copier le contenu d'un fichier sur l'imprimante. Vérifiez que votre imprimante est correctement connectée et prête à imprimer, et tapez :

```
A>PIP LST:=TEXTE.TXT
```

L'imprimante se met en route et votre fichier est imprimé. Vous remarquerez que la syntaxe de cette commande est identique à celle qui est utilisée pour copier un fichier sur une autre disquette, le nom de l'unité de disquettes étant simplement remplacé par LST : qui désigne l'unité d'impression de CP/M.

EFFACEMENT D'UN FICHIER

Il est souvent nécessaire d'effacer les fichiers qui ne sont plus utiles pour faire de la place sur une disquette. On utilise pour cela la commande résidente ERA. Effaçons le fichier TEXTE.TXT en tapant :

```
A>ERA TEXTE.TXT
```

Vérifiez à l'aide de la commande DIR que le fichier a bien disparu du répertoire de la disquette. Il est possible d'utiliser des

caractères génériques avec la commande ERA. Ainsi

```
A>ERA *.TXT
```

effacera tous les fichiers ayant l'extension .TXT et se trouvant sur la disquette par défaut, c'est-à-dire ici celle qui se trouve dans l'unité A, comme l'indique le message A>. Il est indispensable de faire très attention en utilisant cette commande, qui peut s'avérer très dangereuse avec des caractères génériques. Il sera souvent préférable d'effacer les fichiers un par un.

COPIE D'UNE DISQUETTE

Nous avons vu au Chapitre 1 une méthode permettant de copier une disquette sur un système à une seule unité de disquettes. Il existe une méthode plus rapide pour copier des disquettes sur un système à deux unités en utilisant la commande COPYDISC, qui sera décrite au Chapitre 3. La copie exécutée à l'aide de la commande DISCCOPY ou de la commande COPYDISC est exactement identique à l'original. Cette méthode a l'avantage d'être rapide. Elle présente cependant un inconvénient majeur, lié à la façon dont les fichiers sont organisés sur une disquette CP/M. Les disquettes sont divisées en pistes circulaires et en secteurs.

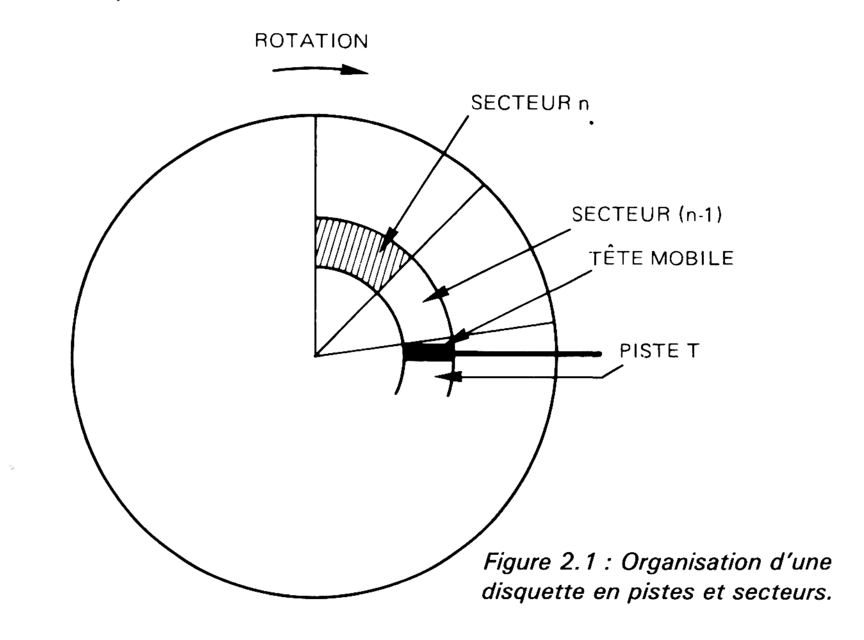


Figure 2.1 : Organisation d'une disquette en pistes et secteurs.

Lorsqu'un fichier est créé sur une disquette, il arrive que l'espace libre ne soit pas d'un seul tenant. Le fichier créé occupera alors les premiers secteurs libres et, si cela est insuffisant, la suite du fichier se trouvera à un autre endroit de la disquette. A force de manipulations (copie de fichiers, effacement, modification), on peut se trouver avec des fichiers éclatés en plusieurs morceaux, ce qui a pour effet de ralentir les opérations de lecture. (Le répertoire contient toutes les informations nécessaires pour gérer ces fichiers éclatés.) Il est alors souhaitable de remettre un peu d'ordre. Pour cela, il suffit d'effectuer une copie fichier par fichier de la disquette en appliquant la méthode suivante.

Placez la disquette à copier dans l'unité A et une disquette vierge formatée dans l'unité B. (Contrairement à la méthode précédente, la disquette devant recevoir la copie doit impérativement être vierge et formatée.) Tapez ↑C pour que l'écriture sur la disquette B soit possible, puis :

```
A>PIP B:=A:*.*
```

Cette commande effectue une copie de chaque fichier se trouvant sur la disquette de l'unité A et la place sur la disquette B. Les fichiers étant copiés un par un sur une disquette vierge, ils occuperont des secteurs contigus. Pour que cette méthode soit utilisable, il faut que le programme PIP.COM se trouve sur la disquette à copier. Si ce n'est pas le cas, il faudra procéder en deux étapes : d'abord charger la commande PIP en mémoire, puis changer de disquette et exécuter la commande.

Placez la disquette système ou une disquette contenant le programme PIP.COM dans l'unité A et la disquette devant recevoir la copie dans l'unité B. Tapez ↑C pour rendre l'écriture possible, puis :

```
A>PIP
```

La commande PIP est chargée en mémoire et l'écran affiche :

```
*
```

qui est l'indicatif de PIP. Sortez la disquette de l'unité A et placez-y la disquette à copier. *Ne tapez pas* ↑C, ce qui aurait pour effet de terminer la commande PIP et de retourner à CP/M. (↑C n'est pas

nécessaire malgré le changement de disquette dans l'unité A, car celle-ci ne sera utilisée qu'en lecture.) Tapez :

```
B:=A:*.*
```

La copie est effectuée comme précédemment.

Ce chapitre a introduit les principales commandes de CP/M. Le Chapitre 3 les présentera plus en détail sous forme de répertoire alphabétique.

3

LES COMMANDES DE CP/M

Dans ce chapitre, toutes les commandes de CP/M disponibles sur l'Amstrad seront présentées par ordre alphabétique et décrites en détail, à l'exception des commandes PIP, ED, ASM et DDT à chacune desquelles un chapitre séparé sera consacré.

La présentation suivante a été adoptée :

- Les mots en MAJUSCULES représentent des mots clés qui doivent être tapés tels quels. Ils peuvent cependant être tapés en minuscules.
- Les mots en <minuscules entre crochets pointus> représentent des paramètres qui doivent être fournis par l'utilisateur. Ainsi <nom de fichier> indique qu'un nom de fichier doit être tapé.
- Tous les signes spéciaux autres que les crochets <et> doivent être tapés.



Fonction

Permet de quitter CP/M pour retourner sous AMSDOS (c'est l'inverse de la commande AMSDOS ¦CPM).

Type

Non résidente.

Syntaxe

AMSDOS

Utilisation

Pour retourner à AMSDOS et au BASIC, tapez :

```
A>AMSDOS
```

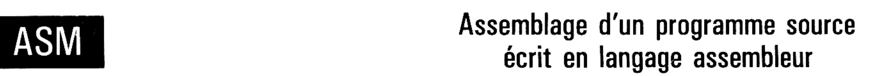
L'écran affiche le message suivant :

```
Amstrad 64K Microcomputer (V2)
(c) 1984 Amstrad Consumer Electronics plc
         and Locomotive Software Ltd.

BASIC 1.1

Ready
```

L'ordinateur est prêt à accepter une commande BASIC ou AMSDOS.



Cette commande est décrite au Chapitre 6.



Fonction

Cette commande permet d'effectuer une copie du secteur de chargement d'une disquette (piste 0 secteur 41 ; les pistes sont numérotées de 0 à 39 et les secteurs de 41 à 49) et du secteur de configuration (piste 0 secteur 42).

Type

Non résidente.

Syntaxe

BOOTGEN

Utilisation

La commande BOOTGEN ne nécessite pas de paramètres. La copie est effectuée d'une disquette sur une autre et des messages sont affichés demandant l'introduction de la disquette source et de la disquette destination. Après la frappe de la commande, l'écran affiche :

```
BOOTGEN V2.0

Please insert SOURCE disc into drive A then press any key:
```

Placez la disquette contenant la configuration à copier dans le lecteur A et tapez une touche quelconque. L'écran affiche alors :

```
Please insert DESTINATION disc into drive A then press any key:
```

Placez la disquette devant recevoir la copie dans l'unité A et tapez une touche quelconque. Après la copie, la possibilité vous est offerte de configurer d'autres disquettes :

```
Do you wish to reconfigure another disc (Y/N):
```

Si vous répondez oui (Y), le cycle recommence. Dans le cas

contraire, l'ordinateur vous demande d'introduire dans le lecteur A une disquette système et de frapper une touche quelconque :

Please insert a CP/M system disc into drive A then press any key:

Le programme se termine alors et un démarrage à chaud est effectué.

Note :

Cette commande est nécessaire pour copier sur plusieurs disquettes une configuration mise au point à l'aide de la commande SETUP (voir cette commande).

Comparaison de deux disquettes
sur un système à deux lecteurs



Fonction

Vérifier que deux disquettes sont physiquement identiques.

Type

Non résidente.

Syntaxe

CHKDISC.

Utilisation

Cette commande ne nécessite pas de paramètres. Le message :

Please insert source disc into drive A and destination disc into drive B then press any key:

demande l'introduction des disquettes à comparer dans les lecteurs. En cas de différences, le message :

WARNING: Failed to compare disc correctly

est affiché. Si les disquettes sont identiques, l'ordinateur affiche le message suivant :

Copy checking complete

puis propose une nouvelle comparaison :

Do you want to check another disc (Y/N):

Répondez Y s'il y a d'autres disquettes à comparer, N dans le cas contraire.



Chargement d'un fichier sur cassette

Fonction

Transférer un fichier d'une cassette vers une disquette.

Type

Non résidente.

Syntaxe

CLOAD "<nom de fichier>" <nom de fichier>
CLOAD "<nom de fichier>"
CLOAD <nom de fichier>
CLOAD

Utilisation

La commande CLOAD prend pour paramètres éventuels deux noms de fichiers. Le premier nom de fichier est le nom utilisé sur la cassette. Il peut être précédé d'un point d'exclamation. Le second nom de fichier est le nom utilisé sur la disquette. Si le premier nom est omis, le premier fichier rencontré sur la bande est chargé. Si le second nom est omis, le fichier chargé sera sauvegardé sur la disquette sous le nom qu'il portait sur la cassette. Ce nom doit bien sûr être correct syntaxiquement pour CP/M. Pour charger par exemple le fichier PROG DEMO de la cassette sur la disquette, tapez :

```
A>CLOAD "PROG DEMO" PROG.DEM
```

Le programme affiche alors :

```
CLOAD V2.0

Press PLAY then any key
```

Enfoncez la touche PLAY du magnétophone après avoir rembobiné la bande. Le début du chargement est signalé par le message :

```
Loading PROG DEMO block 1
```

Si le premier caractère du nom du fichier cassette est un point d'exclamation, les messages seront supprimés et le chargement commencera dès que vous aurez frappé la touche ENTER. Dans ce cas, assurez-vous préalablement que la bande est correctement positionnée et que la touche PLAY est enfoncée.

Note :

La lecture des fichiers sur cassettes peut se faire à 1 000 bps ou à 2 000 bps. L'ordinateur choisit la vitesse en fonction de l'option utilisée lors de la sauvegarde du fichier (voir la commande CSAVE).



Fonction

Copie d'une disquette sur un système à deux unités.

Syntaxe

COPYDISC

Utilisation

Placez une disquette contenant le programme COPYDISC.COM dans l'unité A et tapez :

```
A>COPYDISC
```

L'ordinateur affiche :

```
Please insert source disc into drive A and destination disc into drive B then press any key:
```

Placez la disquette à copier dans l'unité A et la disquette devant recevoir la copie dans l'unité B et tapez une touche quelconque. Une fois la copie terminée, le message :

```
Do you want to copy another disc (Y/N):
```

est affiché. Répondez Y si vous voulez copier d'autres disquettes ou N dans le cas contraire. L'ordinateur affiche alors un nouveau message :

```
Please insert a CP/M system disc into drive A then press any key:
```

Placez une disquette système CP/M dans l'unité A et tapez une touche quelconque.

Note :

La disquette recevant la copie n'a pas besoin d'être formatée. Si cette disquette contient déjà des informations, celles-ci seront effa-

cées dans l'opération. Si une erreur se produit pendant la copie, l'ordinateur affiche un des messages suivants :

Failed to read source disc correctly: track x sector y

Failed to write destination disc correctly: track x sector y

Failed to read destination disc correctly: track x sector y

Failed to verify destination disc correctly: track x sector y

Failed to format destination disc correctly: track x sector y

Dans le premier cas, la disquette originale n'a pu être lue. Elle est probablement endommagée. Les quatre messages suivants indiquent qu'une erreur s'est produite sur la disquette destination. Recommencez l'opération avec une autre disquette.



Sauvegarde d'un fichier sur cassette

Fonction

Transférer un fichier d'une disquette vers une cassette.

Type

Non résidente.

Syntaxe

CSAVE <nom de fichier> "<nom de fichier>" n
CSAVE <nom de fichier>

Utilisation

La commande CSAVE prend pour paramètres éventuels deux noms de fichiers. Le premier nom de fichier est le nom utilisé sur la disquette. Le second nom de fichier est le nom utilisé sur la cassette. Il peut être précédé d'un point d'exclamation. Si le second nom est omis, le fichier sera sauvegardé sur la cassette sous le nom qu'il portait sur la disquette. Pour sauvegarder par exemple le fichier PROG.DEM sur une cassette, tapez :

```
A>CSAVE PROG.DEM
```

L'écran affiche alors :

```
CSAVE V2.0

Press REC and PLAY then any key
```

Enfoncez la touche PLAY et la touche REC du magnétophone simultanément. Le début de la sauvegarde est signalé par le message :

```
Saving PROG.DEM block 1
```

Si un nom de fichier cassette est utilisé et si le premier caractère en est un point d'exclamation (!), les messages seront supprimés et

la sauvegarde commencera dès que vous aurez frappé la touche ENTER. Dans ce cas, assurez-vous préalablement que la bande est correctement positionnée et que les touches PLAY et REC du magnétophone sont enfoncées. L'option ! permet de spécifier la vitesse d'enregistrement des données sur la bande grâce au paramètre n. Ce paramètre peut prendre la valeur 0 qui correspond à une vitesse de 1 000 bps ou la valeur 1 qui correspond à 2 000 bps. Exemple :

```
A>CSAVE PROG.DEM "!PROG DEMO" 1
```

Cette commande entraîne la sauvegarde du fichier PROG.DEM situé sur la disquette se trouvant dans l'unité par défaut (ici l'unité A). Ce fichier est sauvegardé sur la cassette à la vitesse de 2 000 bps et sous le nom PROG DEMO. Le programme n'affiche aucun message et la sauvegarde commence dès la frappe de la touche ENTER.

Note :

La vitesse de sauvegarde par défaut est 1 000 bps. Lors du chargement d'un fichier avec la commande CLOAD, la vitesse de chargement est automatiquement adaptée à la vitesse utilisée pour la sauvegarde.



Mise au point d'un programme ou d'un fichier

Cette commande est traitée au Chapitre 7.

Affichage du répertoire d'une disquette



Fonction

Afficher sur l'écran le répertoire ou une partie du répertoire d'une disquette.

Type

Résidente.

Syntaxe

DIR
DIR <nom d'unité :>
DIR <nom de fichier>

Utilisation

En l'absence de tout paramètre, la commande DIR affiche sur l'écran la liste des fichiers présents dans la zone utilisateur active de la disquette se trouvant dans l'unité par défaut (pour tout renseignement concernant les zones utilisateur, voir la commande USER). Seuls les fichiers possédant l'attribut $DIR seront affichés. Les fichiers possédant l'attribut $SYS n'apparaîtront pas (voir la commande STAT). Si un nom d'unité de disquettes est fourni comme paramètre, le répertoire de la disquette se trouvant dans l'unité spécifiée sera affiché. Si un nom de fichier est fourni, seul ce fichier sera affiché s'il est sur la disquette se trouvant dans l'unité active et dans la zone utilisateur courante. Dans le cas contraire, le message :

```
NO FILE
```

sera affiché. Le nom de fichier peut également être précédé de l'indication d'une unité de disquettes si celle-ci est différente de l'unité par défaut. Ainsi, la commande :

```
A>DIR B:LETTRE. TXT
```

affichera :

```
B:LETTRE .TXT
```

si le fichier LETTRE.TXT est sur la disquette se trouvant dans l'unité B et dans la zone utilisateur courante, et :

```
NO FILE
```

dans le cas contraire.

Des caractères génériques peuvent également être utilisés avec la commande DIR (pour plus d'information sur l'utilisation des caractères génériques, voir le Chapitre 2). Exemple :

```
A>DIR *.COM
```

affichera la liste des fichiers ayant l'extension .COM et se trouvant sur la disquette présente dans l'unité par défaut (ici, l'unité A) et dans la zone utilisateur courante. La commande :

```
A>DIR B:LETTRE.?XT
```

affichera la liste de tous les fichiers se trouvant sur la disquette présente dans l'unité B et dans la zone utilisateur courante, et ayant le nom LETTRE et une extension se terminant par XT.

Note :

Pour obtenir une copie imprimée du répertoire d'une disquette, tapez ↑P avant de lancer la commande. Le répertoire sera affiché en même temps sur l'écran et sur l'imprimante. N'oubliez pas de taper de nouveau ↑P après l'impression du répertoire pour supprimer l'écho sur l'imprimante.

Copie d'une disquette



Fonction

La commande DISCCOPY permet de copier une disquette sur un système à un seul lecteur.

Type

Non résidente.

Syntaxe

DISCCOPY

Utilisation

Pour copier une disquette sur un système à un seul lecteur, procédez de la façon suivante. Placez dans le lecteur une disquette contenant le programme DISCCOPY.COM et tapez :

```
A>DISCCOPY
```

L'ordinateur affiche :

```
DISCCOPY V2.0

Please insert source disc into drive A then press any key:_
```

Ce message vous demande d'introduire dans le lecteur A la disquette à copier (disquette source) puis de taper une touche quelconque. Un nouveau message s'affiche pour indiquer le début de la copie et l'ordinateur lit les huit premières pistes de la disquette et affiche un nouveau message :

```
Please insert destination disc into drive A then press any key:_
```

Ce message vous demande d'introduire la disquette devant recevoir la copie (disquette destination) dans le lecteur A et de taper une touche quelconque. L'ordinateur écrit alors les huit premières pistes et l'opération est répétée cinq fois pour copier la disquette

entière. Si la disquette destination n'est pas formatée, le programme lui donne le même format que la disquette source tout en affichant le message :

Formatting whilst copying

Une fois la copie terminée, le message suivant est affiché :

Copying complete
Do you want to copy another disc (Y/N):_

Répondez Y si vous voulez copier une autre disquette ou N dans le cas contraire.
Si, lors de la copie, vous vous trompez de disquette, un message d'erreur du type :

You must insert the destination disc into drive A

Please insert the destination disc into drive A then press any key:_

sera affiché, vous permettant de corriger votre erreur. De même, si la disquette devant recevoir la copie (disquette destination) est protégée, le message :

The destination disc in drive A must be write-enabled

Please insert destination disc into drive A then press any key:_

sera affiché et vous pourrez corriger l'erreur. Si par contre vous désirez interrompre l'opération, tapez ↑C. L'écran affiche :

WARNING: Failed to copy disc correctly
The destination disc should not be used until it is successfully copied on to

Please insert a CP/M system disc into drive A then press any key:_

Ce message vous informe que la copie a échoué et que le disque destination ne doit pas être utilisé. Bien sûr, dans le cas présent, le disque peut être utilisé, puisqu'il n'a pas été modifié. Si par contre la disquette destination comporte un défaut empêchant l'écriture, l'ordinateur affiche le message :

Drive A: read fail

Retry, Ignore or Cancel?

Vous avez alors la possibilité d'essayer à nouveau (R), d'ignorer l'erreur (I) ou d'interrompre la copie (C). L'option R produira généralement un nouveau message d'erreur. L'option I permet parfois de terminer la copie, mais celle-ci risque d'être inutilisable. L'ordinateur affichera alors le numéro de la piste et le numéro du secteur défectueux, par exemple :

```
Failed to verify the destination disc correctly: track 4 sector 71
```

A la fin de la copie, le message vu précédemment sera affiché :

```
WARNING: Failed to copy disc correctly
The destination disc should not be used until it is successfully copied on to

Please insert a CP/M system disc into drive A then press any key:_
```

indiquant que la copie ne doit pas être utilisée. Si par contre vous choisissez l'option C, la copie sera immédiatement stoppée avec le même message d'erreur.

Dans tous les cas, l'ordinateur vous demande d'introduire dans le lecteur une disquette système CP/M et de taper une touche quelconque.

Note :

La commande DISCCOPY ne copie qu'une seule face de la disquette.



Fonction

La commande DISCCHK permet, sur un système à un seul lecteur, de vérifier si la copie d'une disquette est bien conforme à l'original.

Type

Non résidente.

Syntaxe

DISCCHK

Utilisation

Après avoir copié une disquette, il est nécessaire de vérifier que la copie est bien conforme à l'original. Pour cela, placez dans le lecteur une disquette contenant le programme DISCCHK.COM et tapez :

```
A>DISCCHK
```

L'écran affiche :

```
DISCCHK V2.0

Please insert source disc into drive A then press any key:_
```

Placez la disquette originale (disquette source) dans le lecteur A et tapez une touche quelconque. L'ordinateur lit les huit premières pistes de la disquette et affiche un nouveau message :

```
Please insert destination disc into drive A then press any key:_
```

Insérez la copie (disquette destination) dans le lecteur A et tapez une touche quelconque. La même opération est répétée cinq fois pour comparer les quarante pistes des deux disquettes. Si les deux

disquettes sont identiques, la comparaison se termine par l'affichage du message suivant :

```
Copy checking complete
Do you want to check another disc (Y/N):_
```

Répondez Y si vous avez d'autres disquettes à comparer ou N dans le cas contraire. Si les deux disquettes ne sont pas identiques, l'ordinateur affiche (par exemple) :

```
Failed to verify the destination disc correctly: track 2 sector 66
```

indiquant le numéro de la piste et du secteur où une différence a été rencontrée. Dans ce cas, la comparaison se termine par le message :

```
Copy checking complete
WARNING: Failed to compare discs correctly
Do you want to check another disc (Y/N):_
```

indiquant que la comparaison a échoué. A tout moment, vous pouvez abandonner la comparaison en tapant ↑C. L'ordinateur vous demande alors d'introduire dans le lecteur une disquette système CP/M en affichant le message :

```
↑C...aborted
Please insert a CP/M disc into drive A then press any key:_
```

Tapez une touche quelconque pour revenir à CP/M. L'interruption du programme fait gagner du temps lorsqu'une erreur est détectée, car de toute façon la copie doit être recommencée.

Note :

La commande DISCCHK ne compare qu'une seule face des disquettes.



DISCKIT2

(CPC 6128 uniquement)
Copie, formatage et vérification de disquette

Fonction

Cette commande, particulière à la version de CP/M 2.2 livrée avec l'Amstrad CPC 6128, permet de copier, formater et vérifier des disquettes sur un système à une ou deux unités. Elle est identique à la commande DISCKIT3 de CP/M plus.

Type

Non résidente.

Syntaxe

DISCKIT2

Utilisation

Une fois la commande DISCKIT2 lancée, l'écran affiche :

```
          DISC KIT 1.0
        CPC6128 &  CP/M 2.2
(c) 1985 Amstrad Consumer Electronic plc
      and Locomotive Software Ltd.
One drive found

Copy                    7
Format                  4
Verify                  1
Exit from program       0
```

Si une unité de disquettes externe est connectée, le message "One drive found" est remplacé par "Two drives found".

1. Utilisation sur un système à une seule unité

Copie de disquettes

Tapez la touche f7. L'écran affiche maintenant :

```
Y     Copy
      Any other key to exit to menu
```

Placez la disquette à copier dans l'unité et tapez Y pour commencer la copie (toute autre touche vous ramène au menu précédent). Un message est affiché indiquant le format de la disquette. Pour accélérer la copie, une partie de la mémoire écran est utilisée, ce qui produit un affichage aléatoire dans la partie basse de l'écran. Un nouveau message demande l'introduction dans l'unité de la disquette devant recevoir la copie (il est important de noter que toutes les informations se trouvant sur la disquette recevant la copie seront effacées) :

```
Insert disc to WRITE
Press any key to continue
```

Introduisez la disquette devant recevoir la copie puis tapez une touche quelconque. Si cette disquette n'est pas formatée, l'écran affiche :

```
Disc isn't formatted (or faulty)
Going to format while copying
Disc will be XXXXX format
```

où XXXXX indique le format de la disquette à copier.

L'ordinateur affiche un nouveau message :

```
Insert disc to READ
Press any key to continue
```

Placez dans l'unité la disquette à copier et tapez une touche quelconque. L'opération est répétée plusieurs fois jusqu'à ce que la copie soit terminée, puis le message suivant est affiché :

```
Copy completed
Remove disc
Press any key to continue
```

Retirez la copie de l'unité de disquettes et tapez une touche quelconque. Vous pouvez alors copier une autre disquette en tapant Y, ou retourner au menu général en tapant une autre touche.

Formatage

Pour formater une disquette, tapez la touche f4. L'écran affiche :

```
System format        9
Data format          6
Vendor format        3
Exit menu            .
```

Tapez f9 pour formater une disquette au format SYSTÈME, f6 pour le format DONNÉES ou f3 pour le format VENDEUR (les différents formats sont décrits à l'Annexe 3). Tapez un point (.) sur le clavier numérique pour retourner au menu principal. Si vous choisissez le format SYSTÈME, l'écran affiche le message :

```
About to read reserved tracks
Insert a system disc
Press any key to continue
```

L'ordinateur doit lire les pistes système sur une disquette pour pouvoir les reproduire sur la disquette à formater. Placez une disquette système dans l'unité et tapez une touche quelconque. L'ordinateur lit les pistes 1 et 2 et affiche :

```
Remove disc
Press any key to continue
```

Retirez la disquette de l'unité et tapez une touche. Dans tous les autres cas, l'ordinateur affiche directement le message suivant :

```
Y    Format as XXXXX
     Any other key to exit menu
```

Tapez Y pour formater au format indiqué par XXXXX. Un nouveau message est affiché :

```
Insert disc to format
Press any key to continue
```

Introduisez la disquette à formater et tapez une touche quelconque. Une fois le formatage terminé, l'écran affiche :

```
Format complete
Remove disc
Press any key to continue
```

Retirez la disquette de l'unité et tapez une touche quelconque. Vous pouvez alors formater une autre disquette en tapant Y ou retourner au menu principal en tapant une autre touche. Si un problème est rencontré lors du formatage, le message suivant est affiché :

```
Disc error track Z, sector #Y
__ XXXXX
R-etry or C-ancel?
```

où Z est le numéro de la piste et Y le numéro du secteur où l'erreur s'est produite, et XXXXX un message dépendant du type d'erreur. Vous pouvez alors ressayer de formater la disquette en tapant R, ou annuler l'opération en tapant C. Dans ce cas, l'ordinateur affiche :

```
Disc not formatted - DO NOT USE
Remove disc
Press any key to continue
```

Retirez la disquette de l'unité et tapez une touche quelconque. La disquette ne doit pas être utilisée, car elle est défectueuse. Si la disquette à formater est protégée, le message :

```
Disc write-protected
Insert disc to format
R-etry or C-ancel
```

est affiché. Vous pouvez alors déprotéger la disquette (voir Chapitre 1) et taper R pour recommencer, ou taper C pour annuler l'opération.

Vérification d'une disquette

Pour vérifier une disquette, insérez-la dans l'unité et tapez f1 puis Y. Le format de la disquette est affiché, suivi du message :

```
Verify completed
Remove disc
Press any key to continue
```

Retirez la disquette vérifiée et tapez une touche quelconque. Si une erreur est rencontrée pendant la vérification, les numéros de piste et de secteur où l'erreur s'est produite sont affichés, ainsi qu'un message indiquant la nature de l'erreur. Vous avez alors trois possibilités :

- taper R pour ressayer de lire la disquette ;
- taper I pour continuer la vérification ;
- taper C pour annuler la commande.

Dans tous les cas, l'ordinateur affiche en fin de vérification le message :

```
Disc not completely verified
Remove disc
Press any key to continue
```

Retirez la disquette et tapez une touche quelconque. Pour retourner sous CP/M, tapez la touche f0 lorsque le menu principal est affiché.

2. Utilisation sur un système à deux unités

Copie de disquettes

Si votre système comporte deux unités de disquettes, le menu de copie est le suivant :

Read from A:	8
Read from B:	5
Exit menu	2

Tapez f8 pour copier à partir de l'unité A, f5 pour copier à partir de l'unité B ou f2 pour retourner au menu principal. Si vous tapez f8 ou f5, l'écran affiche :

Write to A:	9
Write to B:	6
Exit to menu	3

Tapez f9 si la disquette recevant la copie doit se trouver dans l'unité A et f6 si elle doit se trouver dans l'unité B. Tapez ensuite Y pour lancer la copie ou une autre touche pour annuler. La copie

fonctionne de la même façon qu'avec un système à une seule unité, excepté le fait que si l'unité contenant la disquette à copier et celle recevant la copie sont différentes, la copie est exécutée en une seule opération, éliminant la nécessité de changer plusieurs fois de disquette. Attention : il est impératif de bien respecter l'ordre des disquettes. Le message :

```
Copy from A: to B:
```

indique que l'original doit se trouver en A et la copie en B. Le message :

```
Copy from B: to A:
```

indique que l'original doit se trouver en B et la copie en A.

Formatage et vérification de disquettes

Ces deux opérations se déroulent de la même façon que sur un système à une seule unité, mais le choix vous est donné de placer la disquette à formater ou à vérifier dans l'unité A ou dans l'unité B.



DRLKEYS

(CPC 6128 uniquement)
Configuration du clavier et de l'écran

Fonction

Cette commande, particulière à la version de CP/M 2.2 de l'Amstrad CPC 6128, sert à configurer le clavier et l'écran pour l'utilisation de LOGO2.

Type

Non résidente.

Syntaxe

DRLKEYS

Utilisation

La commande DRLKEYS modifie l'action de certaines touches du clavier pour l'utilisation du langage LOGO (en particulier les touches du curseur). Les couleurs du fond et des caractères sont également modifiées et l'écran est mis en mode 40 colonnes. (Cette commande n'est pas nécessaire avec le CPC 464 et le CPC 664, car sur les disquettes livrées avec ces machines, le secteur de configuration contient les modifications nécessaires.) L'action de la commande DRLKEYS peut être en partie annulée (à l'exception des modifications de couleurs) par la commande FWRESET.

Note :

La commande DRLKEYS ne doit pas être utilisée pour d'autres usages que la programmation en LOGO. De même, la commande FWRESET doit être utilisée immédiatement après avoir quitté LOGO. Toute utilisation du clavier modifié en dehors de LOGO peut produire des résultats imprévisibles pouvant aller jusqu'à un "plantage" complet de l'ordinateur. (Il ne vous restera plus alors qu'à éteindre l'ordinateur après avoir retiré la disquette.) Il est plus sûr de lancer LOGO2 en utilisant la commande :

```
A>SUBMIT LOGO2
```

qui effectue automatiquement les modifications nécessaires.



Affichage du contenu d'un fichier binaire

Fonction

La commande DUMP permet d'afficher sur l'écran en hexadécimal le contenu d'un fichier binaire.

Type

Non résidente.

Syntaxe

DUMP <nom de fichier>

Utilisation

Nous allons par exemple visualiser le contenu du fichier STAT.COM. Tapez :

```
A>DUMP STAT.COM
```

L'écran affiche une série de chiffres hexadécimaux. La colonne de gauche indique l'adresse à partir du début du fichier. Les seize colonnes suivantes indiquent le contenu de l'adresse indiquée dans la colonne de gauche et des quinze suivantes. La première colonne de données correspond donc à l'adresse indiquée à gauche, la deuxième colonne à adresse + 1, la troisième à adresse + 2, et la dernière colonne à adresse + FF en hexadécimal, soit adresse + 15 en décimal.

On peut interrompre définitivement l'affichage en tapant une touche quelconque. On peut également le suspendre en tapant ↑S. Lorsque l'affichage est suspendu, la frappe d'une touche quelconque le rétablit.

Si le fichier à lister n'existe pas sur la disquette ou si le nom du fichier est omis, le programme affiche le message d'erreur suivant :

```
NO INPUT FILE PRESENT ON DISC
```

Note :

La commande DUMP n'est pas très pratique à utiliser, surtout

lorsque l'ordinateur est en mode rapide (voir la commande SETUP). Il existe de nombreux programmes permettant de visualiser le contenu d'un fichier binaire ligne par ligne et page par page. Ces programmes offrent en général l'avantage de permettre en plus l'affichage des équivalents ASCII à droite de l'écran. Un tel programme est décrit dans l'ouvrage *CP/M approfondi* de A.R. Miller, publié aux éditions Sybex. Il est également possible de modifier le programme DUMP dont le listing source est fourni sur la disquette système.

Cette commande sera décrite en détail au Chapitre 5.



Fonction

La commande ERA est utilisée pour effacer un fichier ou un groupe de fichiers.

Type

Résidente.

Syntaxe

ERA <nom de fichier>
ERA <nom d'unité>:<nom de fichier>

Utilisation

Pour effacer un fichier, tapez ERA suivi du nom du fichier à détruire (séparé par un espace). Si ce fichier ne se trouve pas sur l'unité de disquettes par défaut, il faut faire précéder le nom du fichier du nom de l'unité sur laquelle il se trouve. Par exemple, la commande :

```
A>ERA LETTRE.TXT
```

effacera le fichier LETTRE.TXT se trouvant sur l'unité A (unité par défaut). La commande :

```
A>ERA B:LETTRE.TXT
```

effacera le fichier LETTRE.TXT se trouvant dans l'unité B, bien que l'unité par défaut soit l'unité A. Il faut évidemment que le fichier à détruire existe et qu'il ne soit pas protégé en écriture (attribut $R/W - voir commande STAT). La disquette contenant le fichier ne doit pas non plus être protégée (voir Chapitre 1). Si le fichier n'existe pas, l'ordinateur répond :

```
NO FILE
```

Si le fichier existe mais qu'il est protégé (attribut $R/O), le message suivant est affiché :

```
Bdos Err On A: File R/O
```

et il faut alors taper une touche quelconque pour retourner à CP/M. Si la disquette est protégée en écriture, l'ordinateur affiche le message suivant :

```
Drive A: disc is write protected

Retry, Ignore, Cancel
```

indiquant à l'utilisateur que la disquette est protégée et lui offrant trois possibilités : ressayer (R), ignorer l'erreur (I), ou annuler la commande (C). Dans le premier cas, il est nécessaire de déprotéger la disquette avant de ressayer, sous peine de voir l'erreur se reproduire.

L'option C permet d'annuler la commande en préservant le fichier. L'ordinateur affiche alors le message :

```
Bdos Err On A: Bad Sector
```

et attend la frappe d'une touche quelconque pour repasser le contrôle à CP/M. Ce message est quelque peu inapproprié et ne signifie en rien qu'un secteur de la disquette comporte la moindre défectuosité.

Il est possible d'effacer plusieurs fichiers à l'aide d'une seule commande en utilisant les caractères génériques ? et *. La commande :

```
A>ERA CHAP?.TXT
```

efface tous les fichiers de l'unité par défaut (ici l'unité A) dont le nom comporte quatre caractères commençant par CHAP et dont l'extension est .TXT. La commande :

```
A>ERA B:*.BAS
```

efface tous les fichiers de l'unité B comportant l'extension .BAS. Il est possible d'effacer tous les fichiers se trouvant sur une disquette en utilisant la commande :

```
A>ERA *.*
```

Dans ce cas, l'ordinateur demande confirmation en affichant le message :

```
ALL (Y/N)?
```

Tapez Y suivi de <ENTER> pour confirmer la commande ou N <ENTER> pour annuler.



Fonction

La commande FILECOPY permet de copier un fichier d'une disquette sur une autre ou d'une zone utilisateur dans une autre. Elle permet également de copier un ensemble de fichiers en demandant une confirmation pour chacun d'eux.

Type

Non résidente.

Syntaxe

FILECOPY <nom de fichier>
FILECOPY <nom de fichier> /S<numéro de zone>
FILECOPY <nom de fichier> /D<numéro de zone>

Utilisation

La commande FILECOPY est la seule qui permette de copier des fichiers d'une disquette sur une autre avec un système à un seul lecteur. Pour cela, placez dans le lecteur une disquette contenant le programme FILECOPY.COM et tapez (par exemple) :

```
A>FILECOPY STAT.COM
```

Le programme est chargé en mémoire et l'ordinateur affiche le message suivant :

```
FILECOPY V2.1

Please insert SOURCE disc into drive A then press any key:_
```

Placez la disquette contenant le fichier à copier (ici le fichier STAT.COM) dans le lecteur A et tapez une touche quelconque. Après quelques instants, l'ordinateur affiche :

```
Copying started...

Please insert DESTINATION disc into drive A then press any key:_
```

Placez la disquette devant recevoir la copie du fichier dans le lecteur A et tapez une touche quelconque. L'ordinateur signale que la copie est terminée en affichant :

```
STAT .COM Copied

Copying complete
Please insert a CP/M system disc into drive A then press any key:_
```

Placez une disquette système CP/M dans le lecteur A et tapez une touche quelconque.

La commande FILECOPY permet également de copier une série de fichiers en utilisant les caractères génériques. Nous allons copier tous les fichiers de la disquette système ayant l'extension .COM. Tapez :

```
A>FILECOPY *.COM
```

Après avoir demandé comme précédemment l'introduction de la disquette source et la frappe d'une touche quelconque, le message suivant est affiché :

```
Ambiguous file name: Confirm individual files (Y/N) ?
```

Si vous répondez N à cette question, tous les fichiers comportant l'extension .COM seront copiés. Vous devrez éventuellement changer plusieurs fois de disquette en suivant les messages vous demandant d'introduire alternativement la disquette source et la disquette destination dans le lecteur A. Si vous répondez Y, l'ordinateur vous demandera, pour chaque fichier rencontré, si vous désirez le copier, à l'aide d'un message tel que :

```
MOVCPM .COM Copy (Y/N) ?
```

Répondez Y pour chaque fichier que vous désirez copier et N dans le cas contraire. Lorsque tous les fichiers répondant aux critères demandés ont été proposés, la copie commence. Pour chaque fichier à copier, l'ordinateur vous demande d'introduire dans le lecteur A la disquette source puis la disquette cible. Une fois la copie terminée, la liste des fichiers copiés est affichée sur l'écran.

La commande FILECOPY permet également de copier des fichiers d'une zone utilisateur dans une autre (pour plus de détails concernant les zones utilisateur, voir la commande USER). Pour

copier le fichier STAT.COM (par exemple) de la zone active vers la zone 2, placez une disquette contenant le fichier FILECOPY.COM dans l'unité A et tapez :

```
A>FILECOPY STAT.COM /D2
```

L'écran affiche :

```
FILECOPY V2.1

Copying will be to user 2

Please insert SOURCE disc into drive A then press any key:_
```

Placez une disquette contenant le fichier à copier (ici STAT.COM) dans l'unité A. Le fichier à copier doit être dans la zone active. Tapez une touche quelconque. Le fichier est chargé en mémoire et un nouveau message est affiché :

```
Copying started...

Please insert DESTINATION disc into drive A then press any key:_
```

Placez la disquette devant recevoir la copie dans l'unité A et tapez une touche quelconque. L'écran affiche :

```
STAT .COM Copied

Copying complete
Please insert a CP/M system disc into drive A then press any key:_
```

Placez une disquette système CP/M dans l'unité A et tapez une touche quelconque. Vous pouvez vérifier que la copie a été effectuée en tapant la commande :

```
A>USER 2
```

puis la commande :

```
A>DIR
```

Il est également possible, grâce à la commande FILECOPY, de copier des fichiers se trouvant dans une zone utilisateur quelconque pour les placer dans la zone active. Pour copier, par exemple, le

fichier STAT.COM de la zone utilisateur 2 vers la zone active, placez une disquette contenant le fichier FILECOPY.COM dans l'unité A et tapez :

```
A>FILECOPY STAT.COM /S2
```

Le programme FILECOPY.COM est chargé en mémoire et le message suivant est affiché :

```
FILECOPY V2.1

Copying will be from user 2

Please insert SOURCE disc into drive A then press any key:_
```

Placez la disquette contenant le fichier à copier dans l'unité A et tapez une touche quelconque. L'écran affiche :

```
Copying started...

Please insert DESTINATION disc into drive A then press any key:_
```

Placez la disquette devant recevoir la copie dans l'unité A et tapez une touche quelconque. Un nouveau message s'affiche :

```
STAT .COM Copied

Copying complete
Please insert a CP/M system disc into drive A then press any key:_
```

Tapez une touche pour terminer le programme.

(CPC 6128 uniquement)
Configuration du clavier et de l'écran



Fonction

Rétablir les conditions normales de fonctionnement du clavier et de l'écran (à l'exception des couleurs) après l'utilisation de LOGO2.

Type

Non résidente.

Syntaxe

FWRESET

Utilisation

Cette commande doit être utilisée immédiatement après avoir quitté LOGO2 si l'on veut retravailler sous CP/M (voir la commande DRLKEYS).



Fonction

Formatage d'une disquette vierge ou recyclage d'une disquette déjà utilisée.

Type

Non résidente.

Syntaxe

FORMAT
FORMAT S
FORMAT D
FORMAT I
FORMAT V

Paramètres

Le paramètre S est pris par défaut. (FORMAT est équivalent de FORMAT S.) Dans ce cas, la disquette est formatée au format Amstrad et le chargeur, le secteur de configuration et CP/M sont copiés sur les pistes système à partir de la disquette depuis laquelle la commande FORMAT a été lancée. (Voir également les commandes BOOTGEN et SYSGEN.) C'est pratiquement la seule option à utiliser sur un système à un seul lecteur de disquettes. Le paramètre D spécifie le format Données. La totalité de la disquette est utilisée pour le stockage des données. Ni CP/M, ni le chargeur, ni le secteur de configuration ne sont copiés sur la disquette. En conséquence, aucune commande ne peut être lancée à partir de cette disquette, ce qui en limite l'utilisation aux systèmes à deux lecteurs. Le paramètre I donne le même résultat que le paramètre D, à la différence que le format utilisé est le format IBM.

Le paramètre V produit une disquette sur laquelle la place est réservée pour le chargeur, le secteur de configuration et CP/M. Ceux-ci peuvent être ensuite ajoutés sur la disquette à l'aide des commandes SYSGEN et BOOTGEN. Ce format doit être utilisé chaque fois qu'une disquette doit être donnée ou vendue à un tiers,

afin que ce dernier puisse y placer une copie de son CP/M. Il est en effet interdit de donner ou de vendre sa propre copie de CP/M. Toutes les disquettes du commerce contenant des logiciels d'application sont à ce format.

Utilisation

Placez un disque contenant la commande FORMAT dans le lecteur et tapez la commande. Après chargement, l'écran affiche :

Please insert disc to be formatted into drive A then press any key:

Placez le disque à formater dans le lecteur et tapez une touche quelconque. Une fois l'opération terminée, l'ordinateur vous demande si vous voulez formater une autre disquette :

Do you want to format another disc (Y/N):

Répondez Y pour oui ou N pour non. Si vous répondez non, l'écran affiche le message suivant :

Please insert a CP/M system disc into drive A then press any key:

Placez une disquette système CP/M dans le lecteur A et tapez une touche quelconque.

Dans certains cas, la commande FORMAT peut afficher un message d'erreur du type :

Drive A: read fail x

Retry, Ignore or Cancel ?

Ce message signifie qu'une erreur s'est produite sur la piste x. Vous avez le choix entre trois solutions : ressayer (R), ignorer (I) ou stopper (C). Si l'erreur est due à un défaut de la disquette, l'option R donnera le même message. L'option I permet d'ignorer l'erreur et de continuer le formatage. Cependant, aucune gestion des secteurs défectueux n'est effectuée. De plus, il sera possible d'écrire sur ces secteurs défectueux sans qu'aucun message d'erreur soit signalé. Cependant, les données écrites seront irrécupérables. Il est donc *impératif* d'abandonner le formatage en choisissant l'option C et de ne pas utiliser la disquette en cause. (Votre fournisseur de disquettes devra normalement vous l'échanger.)

Note :

Toute disquette neuve doit être formatée avant utilisation. La commande FORMAT peut également être utilisée pour recycler des disquettes ayant contenu des données. Dans ce cas, les données se trouvant sur la disquette seront définitivement effacées.

Il faut noter que la commande FORMAT ne formate qu'une seule face de la disquette.

Attention :

Si on essaie de formater une disquette protégée, l'ordinateur répond par un message d'erreur :

```
The disc to be formatted in drive A must be write-enabled
Please insert disc to be formatted into drive A then press any key:
```

Il est alors possible de changer de disquette, de déprotéger celle qui se trouve dans le lecteur, ou d'abandonner l'opération en tapant ↑C. L'ordinateur demande alors l'introduction d'une disquette système dans le lecteur et la frappe d'une touche quelconque.



Fonction

La commande LOAD permet de créer, à partir d'un fichier .HEX (fichier hexadécimal au format Intel), une image mémoire exécutable, c'est-à-dire un fichier .COM.

Type

Non résidente.

Syntaxe

LOAD <nom de fichier>

Utilisation

La commande LOAD est utilisée pour générer un programme exécutable à partir d'un fichier hexadécimal produit par exemple par l'assembleur ASM. Le fichier à traiter doit posséder l'extension .HEX (les fichiers hexadécimaux produits par l'assembleur portent systématiquement l'extension .HEX). Il n'est pas nécessaire de spécifier l'extension dans la commande. Pour transformer, par exemple, le fichier hexadécimal PROG.HEX en un programme exécutable, tapez :

```
A>LOAD PROG
```

Si l'opération s'est déroulée normalement, elle se termine par un message du type :

```
FIRST ADDRESS    0100
LAST ADDRESS     0212
BYTES READ       0113
RECORDS WRITTEN    03
```

indiquant l'adresse à laquelle le programme sera chargé (0100), l'adresse à laquelle se termine le programme (0212), le nombre d'octets (0113) et le nombre d'enregistrements écrits (03). Si des

codes incorrects sont rencontrés, un message d'erreur est affiché, tel que :

```
INVALID HEX DIGIT
LOAD ADDRESS 0130
ERROR ADDRESS 0140
BYTES READ :
0130 :
0130 : 44 01 CD 72 01 CD 59 01 0F DA 51 01 7C CD 8F 01
```

indiquant l'adresse à laquelle l'erreur s'est produite et affichant les valeurs en cause. Malgré cela, il est possible que le fichier .COM soit créé sur le disque, mais son utilisation peut produire des résultats imprévisibles.

(CPC 464 et CPC 664) **LOGO**

(CPC 6128) **LOGO2**

LOGO n'est pas à proprement parler une commande de CP/M. Cependant, étant donné la structure de CP/M, LOGO fonctionne comme une commande non résidente. (Il en est de même pour tous les programmes d'application tournant sous CP/M.) Il peut donc être lancé en tapant :

A>LOGO (ou LOGO2 sur le CPC 6128)

à condition que le fichier LOGO.COM (ou LOGO2.COM) se trouve dans une unité de disquettes (ici l'unité A). LOGO.COM est la version CP/M 2.2 livrée avec le CPC 464 et le CPC 664. Elle ne peut être utilisée que si CP/M a été chargé à partir de la face 2 de la disquette système. En effet, le secteur de configuration de cette face contient une modification de l'affectation des touches du clavier (voir la commande SETUP). LOGO2.COM est la version CP/M 2.2 du CPC 6128 et se trouve sur la face 4. Le secteur de configuration de cette face ne contient aucune modification du clavier, aussi le chargement de LOGO2 doit-il être précédé de la commande :

A>DRLKEYS

et suivi de la commande :

A>FWRESET

(voir ces deux commandes). Ceci peut être obtenu en utilisant la syntaxe :

A>SUBMIT LOGO2

(le fichier LOGO2.SUB doit alors être présent sur la disquette). La programmation en LOGO ne sera pas étudiée ici, mais l'ouvrage *Amstrad, introduction à LOGO* y est entièrement consacré.



Fonction

CP/M se charge normalement en haut de la mémoire disponible. Il peut être nécessaire de charger CP/M à une adresse plus basse (pour réserver le haut de la mémoire à un autre usage). La commande MOVCPM permet de créer une version de CP/M qui se chargera à une adresse déterminée.

Type

Non résidente.

Syntaxe

MOVCPM <valeur> *
MOVCPM * *

Utilisation

La commande MOVCPM prend pour paramètre une valeur qui indique (en multiples de 256 octets) la taille mémoire allouée à CP/M. Cette valeur doit être comprise entre 64 et 179. Une valeur supérieure n'entraîne pas de message d'erreur, mais l'utilisation de la version générée produira un "plantage" de l'ordinateur. La commande MOVCPM * * permet de générer la version maximale de CP/M (48 K).

Tapez par exemple la commande :

```
A>MOVCPM 130 *
```

L'ordinateur affiche le message suivant :

```
CONSTRUCTING 32k CP/M vers 2.2
READY FOR "SYSGEN"OR
"SAVE 34 CPM32.COM"
```

Notez que le message n'indique que le nombre entier de kilo-octets (32), alors que la taille mémoire réelle est ici de 32,5 K.

La version de CP/M correspondant à cette taille mémoire a été

créée. Elle se trouve pour l'instant en mémoire vive. Elle peut être copiée sur la disquette sous forme d'un fichier en vue d'une utilisation ultérieure, ou directement sur les pistes système d'une disquette. La commande SAVE est utilisée pour écrire la configuration dans un fichier. La commande :

SAVE 34 CPM32.COM

écrira la configuration dans le fichier CPM32.COM (voir la commande SAVE). La commande SYSGEN doit être utilisée pour transcrire la configuration directement sur les pistes système. Cette commande doit alors être utilisée *immédiatement* après la commande MOVCPM avec la syntaxe :

SYSGEN *

(Voir la commande SYSGEN.)

Attention :

Toute omission de l'astérisque à la fin de la commande MOVCPM entraînera immédiatement un "plantage" de l'ordinateur. Dans ce cas, retirez la disquette du lecteur, coupez puis rétablissez l'alimentation et rechargez CP/M. Procédez de la même façon en cas de tentative de chargement d'une version de CP/M générée avec une valeur de paramètre supérieure à 179.

Cette commande sera étudiée au Chapitre 4.

Fonction

Modification du nom d'un fichier.

Type

Résidente.

Syntaxe

REN <nouveau nom> = <ancien nom>
REN <unité>:<nouveau nom> = <unité>:<ancien nom>
(L'unité doit être la même de chaque côté du signe =.)

Utilisation

Pour changer le nom du fichier LETTRE.TXT se trouvant sur l'unité par défaut (A dans cet exemple) en LETTRE.DOC, tapez :

```
A>REN LETTRE.DOC=LETTRE.TXT
```

On peut vérifier, à l'aide de la commande DIR, que le fichier LETTRE.TXT a bien été remplacé par le fichier LETTRE.DOC, identique au premier au nom près. Si le fichier portant l'ancien nom (ici LETTRE.TXT) n'existe pas, l'ordinateur affiche :

```
NO FILE
```

et la commande n'est pas exécutée. S'il existe déjà un fichier portant le nouveau nom, le message :

```
FILE EXISTS
```

est affiché et la commande n'est pas exécutée non plus. Si le fichier à renommer est protégé en écriture (attribut $R/O - voir la commande STAT), le message d'erreur :

```
Bdos Err On A: File R/O
```

est affiché. Cette fois encore, la commande n'est pas exécutée. Il en est de même si la disquette est protégée.

Il est possible de renommer un fichier se trouvant sur une disquette dans une unité autre que celle par défaut en faisant précéder les noms des fichiers par le nom de l'unité. Si le fichier se trouve sur l'unité B, tapez :

```
A>REN B:LETTRE.DOC=B:LETTRE.TXT
```

La même unité doit être mentionnée devant l'ancien et le nouveau noms. La commande :

```
A>REN B:LETTRE.DOC=A:LETTRE.TXT
```

est incorrecte. De même les commandes :

```
A>REN LETTRE.DOC=B:LETTRE.TXT
A>REN B:LETTRE.DOC=LETTRE.TXT
```

sont incorrectes, car un nom de fichier seul est l'équivalent du même nom précédé du nom de l'unité par défaut. Ces deux commandes sont donc équivalentes de :

```
A>REN A:LETTRE.DOC=B:LETTRE.TXT
```

et

```
A>REN B:LETTRE.DOC=A:LETTRE.TXT
```

et produisent un message d'erreur.



Fonction

Sauvegarder une image d'une partie de la mémoire dans un fichier.

Type

Résidente.

Syntaxe

SAVE <nombre de pages> <nom de fichier>

Paramètres

Le nom de fichier peut être précédé de l'indication d'une unité de disquettes si celle-ci doit être différente de l'unité par défaut.

Utilisation

La commande SAVE écrit le contenu de la mémoire à partir de l'adresse 100H dans un fichier spécifié par le paramètre <nom de fichier>. La taille de la zone mémoire à sauvegarder est déterminée par le paramètre <nombre de pages>. Une page correspond à 256 octets. Cette commande est souvent utilisée après la commande MOVCPM (voir cette commande) pour sauvegarder dans un fichier l'image de CP/M générée. Le nombre de pages est alors 34. Une autre utilisation de la commande SAVE est l'écriture sur une disquette d'un programme ou d'un fichier mis au point à l'aide de la commande DDT (cette commande est décrite au Chapitre 7). Le nombre de pages peut alors être calculé de la façon suivante : au chargement du programme par DDT, la valeur affichée pour NEXT est supérieure de 1 à la dernière adresse utilisée. Cette valeur est en hexadécimal. Il faut alors la convertir en décimal, soustraire 256, diviser le résultat obtenu par 256 et arrondir le résultat à l'unité supérieure.



Fonction

Modifier la configuration du clavier, des lecteurs de disquettes, de l'interface série. Définir une séquence d'actions à exécuter après le chargement de CP/M. Mettre à jour le secteur de configuration du disque utilisé.

Type

Non résidente.

Syntaxe

SETUP

Utilisation

Le secteur de configuration de chaque disquette système contient les informations concernant le clavier, les lecteurs de disquettes, l'interface série et une séquence d'actions à accomplir après le chargement de CP/M. La commande SETUP permet de charger ces données en mémoire, de les modifier à l'aide d'un menu et de les recopier sur la disquette après modification. (Voir également la commande BOOTGEN.) Après lancement de la commande, le message suivant est affiché :

```
** Initial command buffer empty

Is this correct (Y/N):
```

ou bien

```
Initial command buffer:

DIR↑M

Is this correct (Y/N):
```

ce qui signifie que la zone des commandes à exécuter à l'initialisation est vide dans le premier cas ou qu'elle contient la commande

DIR (par exemple) dans le second. Si vous voulez modifier cette zone, répondez Y puis tapez la liste des commandes que vous voulez voir exécutées automatiquement après le chargement de CP/M. Ces commandes seront exécutées après un démarrage à froid (mais pas après un démarrage à chaud) exactement comme si elles étaient entrées au clavier. Pour indiquer la frappe de la touche ENTER, tapez ↑M (le caractère ↑ suivi de la lettre M et non pas CTRL-M). Pour terminer l'entrée des commandes, tapez la touche ENTER. Si vous voulez par exemple qu'après chargement de CP/M les commandes DIR et STAT soient exécutées automatiquement, tapez :

```
DIR↑MSTAT↑M <ENTER>
```

L'ordinateur répond :

```
Initial command buffer:

dir↑Mstat↑M

Is this correct (Y/N):
```

Répondez Y pour accepter ou N pour corriger. L'ordinateur vous propose maintenant de modifier le message d'introduction de CP/M :

```
Sign-on string:

↑\@ww↑\a@@↑]wwCP/M 2.2-Amstrad Consumer Electronics plc↑J↑M

Is this correct (Y/N):
```

Ce message sera envoyé à l'écran à chaque démarrage à froid. Il peut comporter du texte et toute combinaison de caractères de contrôle (voir la liste des caractères de contrôle d'écran en Annexe 10). Nous allons à titre d'exemple étudier le message standard et créer un message différent :

```
↑\@ww
```

Le caractère ↑\ permet d'affecter une valeur à une des quatre couleurs utilisables simultanément sur l'Amstrad. Cette commande prend trois paramètres : numéro de la couleur, valeur 1, valeur 2. Chaque couleur prend deux valeurs qui sont utilisées alternative-

ment. Il est conseillé d'utiliser une valeur 1 et une valeur 2 identiques, sous peine d'obtenir un écran clignotant assez fatigant pour les yeux.

Le numéro de la couleur est ici @, caractère de valeur 64 (voir table des codes ASCII en Annexe 1). Les valeurs sont prises modulo 16 pour le numéro de la couleur (reste de la division par 16) et modulo 32 pour les valeurs. @ correspond au code 64, soit à la couleur n° 0. (Ce numéro correspond au numéro utilisé avec la commande INK du BASIC.) La couleur n° 0 est utilisée par défaut pour le fond de l'écran.

La lettre w a pour code 119 et correspond donc à la valeur 23, soit à la couleur turquoise pastel (la table des couleurs est donnée à l'Annexe 12). Cette valeur est nommée deux fois pour obtenir un fond turquoise.

```
↑\a@@
```

Cette commande fixe de la même façon la couleur n° 1 (a a pour code 97, et 97 modulo 16 = 1) qui est utilisée par défaut pour le texte. Le signe @ a pour code 64, et 64 modulo 32 = 0, ce qui correspond à la couleur noire.

```
↑]ww
```

Cette commande fixe la couleur de la bordure de l'écran. La couleur choisie est ici turquoise pastel comme précédemment. Le reste du message est une ligne de texte terminée par les caractères ↑J (saut de ligne) et ↑M (retour chariot).

Nous allons maintenant remplacer ce message par le message suivant :

```
CP/M 2.2 - Amstrad Consumer Electronics plc
Version personnalisée
```

affiché en bleu marine sur fond vert et bordure rouge vif. A la question :

```
Is this correct (Y/N):
```

répondez par N. L'ordinateur affiche le message suivant :

```
Enter new sign-on string:
```

Tapez :

↑\@II↑\aAA↑\]FFCP/M 2.2 - Amstrad Consumer Electronics plc ↑J↑M Version personnalisee ↑J↑M

puis répondez Y à la question :

Is this correct (Y/N):

Si vous voulez immédiatement tester le résultat, répondez Y à toutes les autres questions. Il est également possible de positionner le message initial n'importe où sur l'écran en utilisant la commande :

↑\↑_xy

dans laquelle x est le caractère ayant pour code ASCII le numéro de colonne, et y le caractère ayant pour code ASCII le numéro de ligne où le message doit apparaître. C'est l'équivalent de la commande BASIC LOCATE (voir Annexe 10).

Le message suivant est :

**Printer power-up string empty

Is this correct (Y/N):

Il est possible de placer ici une commande d'initialisation qui sera envoyée à l'imprimante au moment du démarrage à froid. Cette commande dépend de l'imprimante utilisée et peut servir par exemple à sélectionner une largeur de caractère ou une qualité d'impression.

Tapez N pour placer une commande ou laissez cette zone vide en tapant Y. Le message suivant est :

No keyboard translations set

Is this correct (Y/N):

Il permet de changer l'affectation des touches du clavier. C'est l'équivalent de la commande BASIC KEY DEF. Si vous tapez Y, l'ordinateur affiche :

```
Enter required command from:_
        A - Add key translation (key number, normal, shift, control)
        D - Delete key translation (key number)
        C - Clear all translations
        F - Finish translations

Command:
```

L'option A permet de modifier l'affectation d'une touche. Elle doit être suivie de l'indication du numéro de la touche (les numéros des touches sont listés à l'Annexe 11) et des codes qui doivent lui être affectés en position normale, majuscule (SHIFT) et contrôle (CTRL). Exemple :

```
Command:_A 67,97,65,1
```

permet d'affecter à la touche Q (n° 67) les caractères a (code 97), A (code 65) et ↑A (code 1).

L'option D permet de supprimer l'affectation d'une touche et de lui rendre son fonctionnement normal. Exemple :

```
Command:_D 67
```

supprime l'affectation effectuée précédemment.

L'option C permet de supprimer toutes les affectations et de revenir à un usage normal du clavier.

L'option F termine les opérations d'affectation. L'écran affiche alors (par exemple) :

```
Keyboard translations:

Key code    Normal    Shift    Control
  67          97        65        1

Is this correct (Y/N):
```

Répondez N si vous voulez encore modifier les affectations ou Y si celles-ci sont correctes. Le message suivant :

```
No keyboard expansions set

Is this correct (Y/N):
```

indique qu'aucune touche de fonction n'est programmée. Trente-deux touches de fonction (0 à 31) peuvent être programmées. Ces trente-deux touches de fonction correspondent aux valeurs des touches 128 à 159. Les valeurs 128 à 140 sont affectées aux touches du clavier numérique (0 à 9 : 128 à 137, point décimal : 138, enter : 139 et ↑enter : 140). Les valeurs 141 à 159 doivent être affectées à des touches quelconques suivant la méthode exposée plus haut pour l'affectation de la touche A. Nous allons par exemple affecter à la touche 0 la commande DIR. Au message précédent, répondez N. L'écran affiche :

```
Enter required command from:_
        A - Add keyboard expansion (token number, expansion string)
        D - Delete keyboard expansion (token number)
        C - Clear all keyboard expansions
        F - Finish keyboard expansions
```

Tapez :

```
Command: _A 128 DIR↑M
Command: _F
```

L'ordinateur affiche :

```
Keyboard expansion strings:

Expansion token                 Expansion string

      0                               dir↑M

Is this correct (Y/N):
```

Répondez maintenant Y à tous les messages jusqu'au rechargement de CP/M pour pouvoir tester le résultat. La touche 0 du clavier numérique permet maintenant d'obtenir immédiatement le répertoire de la disquette se trouvant dans l'unité active. Remarquez qu'il n'est plus possible de se servir du clavier numérique pour obtenir les chiffres. En effet, la valeur 128 est affectée à la touche 0 en usage normal, avec SHIFT et avec CTRL.

Lorsque aucune fonction n'est affectée à cette touche, elle produit le chiffre 0. Dès qu'une fonction lui est affectée, elle produit cette fonction dans les trois positions (normale, SHIFT et CTRL). Il est possible de changer l'affectation de cette touche et de lui affecter les valeurs suivantes (voir plus haut) :

Key code	Normal	Shift	Control
15	48	128	128

On obtiendra ainsi le chiffre 0 en tapant la touche 0 seule et la commande DIR en tapant SHIFT-0 et ↑0. On peut également obtenir deux fonctions différentes, l'une avec SHIFT et l'autre avec CTRL.

Le message suivant est :

```
Default IO byte settings are

    CON: is assigned to CRT : (keyboard and VDU)
    RDR: is assigned to TTY : (special IO device 0)
    PUN: is assigned to TTY : (special IO device 0)
    LST : is assigned to LPT : (centronics printer)

Is this correct (Y/N):
```

CP/M reconnaît, en plus des unités de disques, quatre unités logiques : CON: (console), RDR: (unité de lecture, à l'origine lecteur de bandes papier perforées), PUN: (unité de sortie, à l'origine perforateur de bandes papier) et LST: (unité d'impression). Un certain nombre d'unités physiques peuvent être affectées à ces unités logiques (voir commande STAT et Annexe 7). Si vous désirez modifier les affectations par défaut, répondez N au message affiché. L'ordinateur attend alors une nouvelle affectation. Le cas le plus fréquent est l'utilisation d'une imprimante série. Il faut alors affecter l'unité logique d'impression LST: à l'interface série correspondant à l'unité physique TTY:. Tapez :

```
Enter required IO byte setting:_LST: TTY:
```

La modification est reflétée sur l'écran. Si vous acceptez cette modification et si vous l'enregistrez sur le disque, les sorties imprimante seront dirigées vers l'interface série.

L'option suivante permet de choisir entre les modes lent et rapide. L'écran affiche :

```
Default: slow mode set

Is this correct (Y/N):
```

Si vous répondez N, le mode rapide sera sélectionné. Dans ce

mode, les registres secondaires ne sont pas sauvegardés lorsque CP/M accède à la ROM de l'Amstrad. Il est donc souvent préférable, par sécurité, de choisir le mode lent.

L'option suivante permet de supprimer l'affichage des messages d'erreur. Comme l'indique le message affiché sur l'écran :

```
Default: BIOS messages enabled

Is this correct (Y/N):
```

les messages d'erreur sont normalement affichés. Si vous répondez N à ce message, l'écran affiche :

```
Default: BIOS messages disabled

Is this correct (Y/N):
```

Si vous répondez maintenant Y, les messages ne seront plus affichés.

L'option suivante permet de déterminer si le buffer de commande sera ou non effacé par CP/M. Il n'est pas conseillé, sauf pour les programmeurs avertis, de changer cette option. L'écran affiche :

```
Default: Clear initial command buffer on keyboard input

Is this correct (Y/N):
```

Si vous répondez N à cette question, le message affiché devient :

```
Default: Preserve command buffer on keyboard input

Is this correct (Y/N):
```

Les trois options suivantes concernent les temps d'accès et les délais de mise en route et d'arrêt des unités de disquettes. Leur modification est à déconseiller aux utilisateurs non techniciens.

Les deux dernières options concernent les paramètres du SIO (interface série) et permettent de choisir la vitesse de transmission et de réception, le nombre de bits de données, la parité et le nombre de stop bits, cela pour les canaux A et B. (Pour le canal B, une seule vitesse de transmission et de réception peut être spécifiée. Des vitesses de transmission et de réception différentes sont néces-

saires pour les systèmes de type Télétel.) Les paramètres du canal A sont d'abord affichés sur l'écran (tx signifie transmission et rx signifie réception) :

```
Z80 SIO channel A: 9600 tx baudrate, 9600 rx baudrate, 8 data bits
                                No parity, 1 stop bit

Is this correct (Y/N):
```

Pour changer ces paramètres, répondez N à la question puis tapez les nouvelles valeurs séparées par un espace. Ces valeurs peuvent être :

Vitesse	50, 75, 110, 150, 200, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 9600, 19200
Bits de données	5, 6, 7, 8
Parité	EVEN (paire), ODD (impaire), NONE (aucune)
Stop bits	1, 1.5, 2

Exemple :

```
Enter channel A parameters: 75 1200 7 NONE 2
```

Les paramètres du canal B se règlent de la même façon. Une seule valeur de vitesse doit cependant être indiquée pour la transmission et la réception.

Une fois tous les paramètres entrés, il est possible de les enregistrer sur le disque. Le message suivant est affiché :

```
Do you want to update your system disc (Y/N):
```

Si vous répondez N, les nouveaux paramètres sont abandonnés, le disque n'est pas modifié et le programme se termine. Dans le cas contraire, les paramètres sont enregistrés sur le disque, dans le secteur de configuration (piste 0, secteur 42 - les pistes sont numérotées de 0 à 39 et les secteurs de 41 à 49). Ce secteur pourra ensuite être copié sur d'autres disques en utilisant la commande BOOTGEN. Une possibilité est ensuite offerte de recharger CP/M à partir du disque pour tester la nouvelle configuration :

```
Do you want to restart CP/M (Y/N):
```

En cas de réponse négative, le programme se termine. Le disque est modifié, mais la nouvelle configuration ne sera active que lors du prochain démarrage à froid à partir de ce disque. Dans le cas contraire, un démarrage à froid est exécuté et la nouvelle configuration est immédiatement active.

Affichage et modification des caractéristiques des unités et des fichiers



Fonction

La commande STAT permet d'afficher des informations beaucoup plus complètes que celles qui sont fournies par la commande DIR sur les fichiers ainsi que sur les unités CP/M et sur les disquettes.

Type

Non résidente.

Syntaxe

STAT
STAT <unité de disquettes>:
STAT <unité de disquettes>:=R/O
STAT <nom de fichier>
STAT <nom de fichier> $[R/O]
STAT <nom de fichier> $[R/W]
STAT <nom de fichier> $[DIR]
STAT <nom de fichier> $[SYS]
STAT <unité logique>:=<unité physique>
STAT DEV:
STAT VAL:
STAT USR:
STAT <unité de disquettes>:DSK:

Utilisation

Cette commande est utilisée pour afficher ou modifier diverses caractéristiques. La commande :

```
A>STAT
```

affichera la place disponible et le statut de la disquette se trouvant dans l'unité par défaut (ici, l'unité A) sous la forme :

```
A: R/W, Space: 142k
```

Le statut de la disquette est ici R/W (Read/Write - lecture/écriture), ce qui signifie qu'il est possible d'y écrire des informations. L'espace disponible est de 142 K (142 x 1 024 caractères).

La commande :

```
A>STAT A:=R/O
```

permet de modifier le statut de la disquette se trouvant en A. Cette disquette se trouve provisoirement protégée et il est impossible d'y écrire des informations (attention : cette protection est inefficace contre la commande FORMAT ou les commandes DISCCOPY et COPYDISC). Si l'on tape maintenant :

```
A>STAT
```

on obtiendra le message suivant :

```
A:R/O, Space: 142k
```

Pour autoriser de nouveau l'écriture sur la disquette, il suffit de taper :

```
A>STAT A:=R/W
```

La commande STAT permet également d'afficher la taille d'un fichier :

```
A>STAT SYSGEN.COM
```

fournira le message suivant :

```
Recs Bytes Ext Acc
 12     2k   1 R/W A:SYSGEN.COM

Bytes Remaining on A: 142k
```

Ce message indique qu'il reste 142 K libres sur la disquette placée dans l'unité A et que le fichier SYSGEN.COM s'y trouve bien ; qu'il n'est pas protégé contre l'écriture (R/W) et peut donc être effacé ; qu'il est constitué de 12 enregistrements de 128 octets et occupe au total 2 K ; enfin qu'il est référencé par une extension physique. Si le

fichier n'était pas présent sur la disquette, la commande STAT afficherait simplement :

```
File Not Found
```

Une autre option de la commande STAT peut être utilisée pour modifier le statut d'un fichier. Un fichier peut avoir les statuts suivants :

R/O Protégé contre l'écriture. Le fichier ne peut pas être effacé.
R/W Non protégé. Le fichier peut être effacé.
DIR Le nom du fichier apparaîtra dans le répertoire.
SYS Le nom du fichier n'apparaîtra pas dans le répertoire.

Cette option s'utilise de la façon suivante (par exemple) :

```
A>STAT SYSGEN.COM $SYS
```

L'ordinateur répond en affichant le message :

```
SYSGEN.COM set to SYS
```

Le nom du fichier n'apparaît plus dans le répertoire affiché par la commande DIR. Cependant, la commande :

```
A>STAT SYSGEN.COM
```

affichera le message suivant :

```
Recs Bytes Ext Acc
 12     2k   1 R/W A:SYSGEN.COM
Bytes Remaining On A: 142k
```

indiquant que le fichier SYSGEN.COM se trouve toujours sur la disquette. Son statut (SYS) est indiqué par les parenthèses qui entourent son nom. Redonnez à ce fichier son statut initial en tapant :

```
A>STAT SYSGEN.COM $DIR
```

L'option suivante permet de modifier l'affectation des unités logiques et physiques de CP/M. L'affectation par défaut est fixée à chaque démarrage à froid de CP/M et peut être modifiée grâce à la

commande SETUP (voir cette commande). Les unités logiques sont au nombre de quatre :

CON: Unité d'entrée/sortie (console).
RDR: Unité d'entrée.
PUN: Unité de sortie.
LST: Unité de sortie (listage).

RDR et PUN désignent respectivement le lecteur de bandes perforées et le perforateur de bandes, qui étaient des périphériques répandus à l'origine de CP/M.)

Les unités physiques sont :

CRT: Clavier/écran.
TTY: Première unité d'entrée/sortie (interface série).
BAT: Équivaut en entrée à RDR: et en sortie à LST:.
UC1: Seconde unité d'entrée/sortie.
UR1: Seconde unité d'entrée/sortie.
UR2: Clavier.
PTP: Unité de sortie factice (utilisable pour tester des programmes).
PTR: Envoie un caractère "fin de texte" (↑Z).
UP1: Seconde unité d'entrée/sortie.
UP2: Écran.
LPT: Interface parallèle (sortie).
UL1: Seconde unité d'entrée/sortie.

Les affectations possibles sont les suivantes :

CON:	→TTY :	CRT :	BAT :	UC1
RDR:	→TTY :	PTR :	UR1 :	UR2
PUN:	→TTY :	PTP :	UP1 :	UP2
LST:	→TTY :	CRT :	LPT :	UL1

L'affectation usuelle est :

CON: →CRT:
RDR: →TTY:
PUN: →TTY:
LST: →LPT:

Les périphériques de l'Amstrad correspondent aux unités suivantes :

Écran	CRT: UP2:
Clavier	CRT: UR2:
Interface parallèle	LPT:
1ere unité d'entrée/sortie	TTY:
2de unité d'entrée/sortie	UC1: UL1: UP1: UR1:

Les modifications d'affectation les plus fréquentes sont :

LST: →TTY:	Utilisation d'une imprimante connectée à l'interface série.
LST: →CRT:	Envoie des sorties imprimante vers l'écran.
CON: →TTY:	Utilisation d'une console reliée à l'interface série.

La commande :

```
A>STAT DEV:
```

permet de connaître les affectations en usage en affichant (par exemple) :

```
CON: is CRT:
RDR: is TTY:
PUN: is TTY:
LST: is LPT:
```

Pour changer une affectation, tapez par exemple :

```
A>STAT LST:=TTY:
```

ce qui aura pour effet d'envoyer toutes les sorties imprimante vers l'interface série. Vérifiez que la modification a été enregistrée en tapant :

```
A>STAT DEV:
```

ce qui entraîne l'affichage du message :

```
CON: is CRT:
RDR: is TTY:
PUN: is TTY:
LST: is TTY:
```

La commande STAT permet d'afficher des renseignements sur elle-même et ses différents paramètres utilisables ; ainsi :

```
A>STAT VAL:
```

produit l'affichage d'un résumé des paramètres possibles :

```
Temp R/O Disk: d:=R/O
Set indicator  : d:filename.typ$R/O$R/W$SYS$DIR
Disk status    : DSK: d:DSK:
User status    : USR:
Iobyte Assign  :
CON: =TTY: CRT: BAT: UC1
RDR: =TTY: PTR: UR1: UR2
PUN: =TTY: PTP: UP1: UP2
LST:  =TTY: CRP: LPT: UL1
```

Une autre fonction de la commande STAT est d'indiquer la zone utilisateur active et les zones contenant des fichiers (pour tout renseignement concernant les zones utilisateur, voir la commande USER). La commande :

```
A>STAT USR:
```

affichera par exemple :

```
Active user :0
Active files :0  1  7
```

indiquant que la zone active est la zone 0 et que les zones 0, 1 et 7 contiennent des fichiers.

La dernière possibilité offerte par la commande STAT est l'affichage de toutes les caractéristiques physiques d'une disquette. Tapez par exemple :

```
A>STAT DSK:
```

pour obtenir l'affichage des caractéristiques de l'unité de disquettes par défaut (ici, l'unité A). L'écran affiche :

```
A    : Drive Characteristics
1368 : 128 Byte Record Capacity
171  : Kilobyte Drive Capacity
64   : 32 Byte Directory Entries
64   : Checked Directory Entries
128  : Records/Extent
8    : Records/Block
36   : Sectors/Track
2    : Reserved Tracks
```

A	:	Nom de l'unité de disquettes.
1368	:	Nombre d'enregistrements de 128 octets disponibles sur une disquette.
171	:	Capacité d'une disquette en kilo-octets.
64	:	Nombre d'entrées existantes dans le répertoire. Chaque entrée fait 32 octets.
64	:	Nombre d'entrées vérifiées. C'est aussi le nombre maximal de fichiers pouvant se trouver sur une disquette.
128	:	Nombre d'enregistrements adressables par une entrée (extension).
8	:	Nombre d'enregistrements par bloc.
36	:	Nombre de blocs par piste.
2	:	Nombre de pistes réservées pour le système.

Pour obtenir les mêmes renseignements au sujet de l'unité de disquettes B, on utilisera la commande :

```
A>STAT B:DSK:
```

Note :

La commande STAT étant non résidente, elle doit être sur une disquette se trouvant dans une des unités pour pouvoir être employée. Si elle ne se trouve pas sur l'unité par défaut, on peut faire précéder la commande du nom de l'unité contenant la disquette sur laquelle elle se trouve. Par exemple, si l'unité active est A et si la commande STAT.COM est sur la disquette se trouvant dans l'unité B,

```
A>B:STAT
```

permettra d'afficher l'espace disponible et le statut de la disquette de l'unité A, bien que celle-ci ne contienne pas la commande STAT.COM.



Traitement d'un lot de commandes

Fonction

La commande SUBMIT permet de traiter des lots de commandes définis sous forme de fichier texte avec éventuellement passage de paramètres au moment de l'exécution.

Type

Non résidente.

Syntaxe

SUBMIT <nom de fich.> <param1> <param2>. . . <param n>

Utilisation

Le fichier fourni comme paramètre à la commande SUBMIT est un fichier texte qui doit se trouver sur l'unité de disquettes active et dont le nom doit comporter l'extension .SUB. Ce fichier doit contenir une liste de commandes valides CP/M et peut être créé à l'aide de l'éditeur ED. Ces commandes peuvent comporter des paramètres fictifs qui seront remplacés lors de l'exécution par les paramètres d'appel de la commande SUBMIT. Les paramètres fictifs doivent être de la forme $1, $2, $3 ...$n. Si le caractère $ doit être placé dans une commande, il doit être doublé ($$). On peut également insérer dans une commande des caractères de contrôle en tapant d'abord une flèche verticale (↑) suivie du caractère correspondant. Ainsi, pour insérer un caractère contrôle A, il faut taper la flèche verticale (↑) suivie du caractère A majuscule, et non les touches CTRL-A.

Si le nombre de paramètres réels et le nombre de paramètres fictifs ne correspondent pas, la commande n'est pas exécutée et un message d'erreur est affiché. L'exécution d'un fichier de commandes .SUB peut être interrompu par la frappe d'une touche quelconque. Le fichier de commandes suivant peut être utilisé par exemple pour assembler, charger et exécuter un programme en une seule opération :

```
ASM $1
LOAD $1
$1
```

Si le nom de ce fichier est RUN.SUB, la commande :

```
A>SUBMIT RUN DEMO
```

assemble le fichier source DEMO.ASM, charge le fichier hexadécimal DEMO.HEX créé par la commande ASM, et exécute le fichier DEMO.COM créé par la commande LOAD. En fait, la commande SUBMIT crée d'abord un fichier $$$.SUB dans lequel les paramètres fictifs sont remplacés par les paramètres réels :

```
ASM DEMO
LOAD DEMO
DEMO
```

C'est ce fichier qui est ensuite exécuté, puis détruit après l'exécution de la dernière commande. La commande SUBMIT peut traiter des fichiers .SUB se trouvant sur une unité de disquettes différente de l'unité active. Exemple :

```
A>SUBMIT B:RUN DEMO
```

Dans ce cas, le fichier $$$.SUB est créé sur la disquette de l'unité B mais n'est pas exécuté. Son exécution ne sera lancée que si la disquette est placée dans l'unité active et qu'un démarrage à chaud est exécuté.

Le fichier .SUB peut contenir comme dernière commande un appel à la commande SUBMIT, avec un nouveau fichier .SUB comme paramètre.



Fonction

Copier le système CP/M sur une disquette. La disquette devant recevoir la copie doit être au format système (formatée avec l'option S ou l'option V). Le système peut être copié à partir d'une disquette (pas de paramètre), d'un fichier (dont le nom doit alors être fourni comme paramètre à la commande) ou du résultat de l'utilisation de la commande MOVCPM (voir cette commande) si celle-ci a été utilisée immédiatement avant (option *).

Type

Non résidente.

Syntaxe

SYSGEN
SYSGEN *
SYSGEN <nom de fichier>

Utilisation

La commande SYSGEN peut être utilisée sans paramètre. La copie est alors effectuée d'une disquette sur une autre et des messages sont affichés, demandant l'introduction de la disquette source et de la disquette destination. Après la frappe de la commande, l'écran affiche :

```
SYSGEN V2.0

Please insert SOURCE disc into drive A then press any key:
```

Placez la disquette contenant le système à copier dans le lecteur A et tapez une touche quelconque. L'écran affiche alors :

```
Please insert DESTINATION disc into drive A then press any key:
```

Placez la disquette devant recevoir la copie dans l'unité A et tapez une touche quelconque. Après la copie, la possibilité vous est

offerte de configurer d'autres disquettes :

Do you wish to reconfigure another disc (Y/N):

Si vous répondez oui (Y), le cycle recommence. Dans le cas contraire, l'ordinateur vous demande d'introduire dans le lecteur A une disquette système et de frapper une touche quelconque :

Please insert a CP/M system disc into drive A then press any key:

Le programme se termine alors et un démarrage à chaud est effectué.

Un nom de fichier contenant une image du système (produite par la commande MOVCPM) peut également être fourni comme paramètre. Le système sera alors copié à partir de ce fichier. Si la commande MOVCPM a été utilisée immédiatement avant la commande SYSGEN, le paramètre * pourra être utilisé pour désigner le fichier produit à cette occasion.

Note :

La commande SYSGEN doit être utilisée pour copier le système CP/M sur les disquettes formatées avec l'option V. C'est le cas des disquettes du commerce contenant des programmes d'application. La commande SYSGEN peut également être utilisée pour remplacer par une version différente le système se trouvant sur une disquette formatée avec l'option S.

Affichage d'un fichier texte



Fonction

La commande TYPE permet d'envoyer à l'écran le contenu d'un fichier. Il est également possible d'obtenir simultanément une copie sur l'imprimante.

Type

Résidente.

Syntaxe

TYPE <nom de fichier>
TYPE <nom d'unité>:<nom de fichier>

Utilisation

Pour lister le contenu du fichier DUMP.ASM se trouvant sur la disquette système (par exemple), tapez :

```
A>TYPE DUMP.ASM
```

Si ce fichier se trouve sur une unité de disquettes différente de l'unité par défaut (ici A), faites précéder le nom du fichier du nom de l'unité dans laquelle il se trouve, par exemple :

```
A>TYPE B:DUMP.ASM
```

Le contenu du fichier est alors envoyé à l'écran. Les tabulations éventuellement contenues dans le fichier sont étendues à huit espaces. L'affichage peut être interrompu définitivement en tapant une touche quelconque. Il peut être suspendu provisoirement (pour permettre la lecture) en tapant ↑S. Lorsque l'affichage est suspendu, il peut être repris en tapant une touche quelconque. La lecture des fichiers à l'aide de la commande TYPE n'est cependant pas très aisée, surtout si l'on choisit à l'initialisation de CP/M le mode rapide (voir la commande SETUP). Il est préférable d'utiliser un programme d'affichage page par page. Un exemple d'un tel pro-

gramme peut être trouvé dans *CP/M approfondi* de A. R. Miller, éditions Sybex. Une copie imprimée du fichier peut être obtenue en faisant précéder la commande par le caractère de contrôle ↑P qui initialise l'écho sur l'imprimante de tout ce qui est affiché sur l'écran. Cet écho peut être supprimé en tapant ↑P une seconde fois. Il est cependant beaucoup plus pratique d'utiliser la commande PIP (décrite au Chapitre 4), qui autorise la mise en page des listings et la numérotation des lignes. L'usage des caractères génériques est interdit avec la commande TYPE.

Attention :

La commande TYPE ne doit être employée qu'avec des fichiers texte (fichiers contenant des caractères ASCII). Son usage avec des fichiers d'un autre type peut produire des résultats imprévisibles, entraînant généralement un "plantage" de CP/M. Pour afficher le contenu des fichiers ne comportant pas uniquement du texte, utilisez la commande DUMP (voir cette commande).

Modification de la zone utilisateur active



Fonction

La commande USER permet la modification de la zone utilisateur active.

Type

Résidente.

Syntaxe

USER <numéro de zone utilisateur>

Utilisation

CP/M permet de partager une disquette en plusieurs zones utilisateur. Chaque zone est indépendante des autres et a son propre répertoire. L'usage des zones utilisateur est indispensable lorsque plusieurs utilisateurs doivent partager une machine munie d'un disque fixe (disque dur) contenant généralement plusieurs millions de caractères. Sur une machine comme l'Amstrad, qui est équipée d'un lecteur de disquettes de 170 000 caractères environ, chaque utilisateur peut avoir ses propres disquettes, ce qui rend l'utilisation des zones à peu près inutile.

Lors du démarrage de CP/M, la zone active est automatiquement la zone 0. Nous allons prendre pour exemple une copie de la disquette système livrée avec l'ordinateur. La commande :

```
A>STAT USR :
```

affiche le message suivant :

```
Active User : 0
Active Files : 0
```

Ce message indique que la zone utilisateur active est la zone 0 et que seule cette zone contient des fichiers. Pour changer de zone utilisateur, tapez par exemple :

```
A>USER 9
```

Tapez maintenant la commande :

```
A>DIR
```

L'ordinateur affiche le message :

```
NO FILE
```

indiquant qu'aucun fichier ne se trouve dans la zone active. Pour copier un fichier dans une zone utilisateur vide, il faut utiliser la commande FILECOPY (voir cette commande). Une autre technique intéressante à connaître, car elle fonctionne sur tous les systèmes CP/M (la commande FILECOPY est propre à l'Amstrad), est l'utilisation de la commande PIP. Cette commande permet de copier un fichier à partir d'une zone utilisateur non active, mais pas vers une zone non active. Il faut donc la copier dans la zone dans laquelle on veut copier des fichiers. Une astuce est ici nécessaire. Nous allons copier le programme PIP.COM en mémoire à partir de la zone 0. Nous passerons ensuite dans la zone utilisateur 9 où nous copierons, grâce à la commande SAVE (qui, elle, est résidente), la zone mémoire contenant le programme PIP.COM sur la disquette. Repassez dans la zone 0 en tapant :

```
A>USER 0
```

Pour charger le programme PIP.COM en mémoire, il faut utiliser la commande non résidente DDT (voir l'étude détaillée de cette commande au Chapitre 7). Tapez :

```
A>DDT PIP.COM
```

L'écran affiche :

```
DDT VERS 2.2
NEXT PC
1E00 0100
_
```

Le message affiché indique que PIP.COM a été chargé en mémoire et occupe 1E00 octets (en hexadécimal), ce qui correspond à 30 pages de 256 octets. Le tiret affiché sous le message est l'indicatif de DDT, signalant que celui-ci est prêt à exécuter une

commande. Nous allons retourner sous CP/M en tapant la commande :

```
-G0
```

(Attention, il s'agit de la lettre G suivie du chiffre 0, et non du mot anglais *GO*.)

Passez maintenant dans la zone utilisateur 9 en tapant :

```
A>USER 9
```

puis sauvegardez trente pages de la mémoire à l'aide de la commande :

```
A>SAVE 30 PIP.COM
```

La commande :

```
A>DIR
```

permet de vérifier que la zone utilisateur contient bien maintenant le seul fichier PIP.COM. On pourrait procéder de la même façon pour copier d'autres fichiers, mais la commande PIP offre une méthode plus rapide (pour plus de détails sur la commande PIP, voir le Chapitre 4). Nous allons copier le programme STAT.COM de la zone 0 dans la zone 9. Tapez :

```
A>PIP STAT.COM=STAT.COM[G0]
```

Attention, cette fois encore il s'agit de la lettre G suivie du chiffre 0. Ce chiffre indique la zone utilisateur dans laquelle la commande PIP doit aller chercher le fichier à copier. Vous pouvez maintenant utiliser la commande STAT à partir de la zone 9. Tapez par exemple :

```
A>STAT USR:
```

L'écran affiche le message suivant :

```
Active User: 9
Active Files: 0 9
```

indiquant que la zone active est la zone 9 et que les zones 0 et 9 contiennent des fichiers. On peut exécuter la même manœuvre pour copier des fichiers dans toutes les zones utilisateur, sous réserve qu'il y ait assez de place sur la disquette. Il y a seize zones utilisateur, numérotées de 0 à 15. Pour récupérer la place prise sur la disquette par l'opération décrite ci-dessus, effacez tous les fichiers de la zone 9 et repassez en zone 0.

Extension de la commande SUBMIT



Fonction

Permettre le passage d'une série de commandes aux programmes appelés par la commande SUBMIT.

Type

Non résidente.

Syntaxe

XSUB

Utilisation

La commande XSUB est utilisée pour permettre le passage d'une série de commandes aux programmes appelés par la commande SUBMIT. La commande XSUB doit être placée au début du fichier .SUB. Elle est alors chargée en mémoire juste au-dessous du CCP et reste active jusqu'au prochain démarrage à froid. Sa présence est signalée à chaque démarrage à chaud par le message :

```
(xsub active)
```

Voici un exemple d'utilisation avec la commande ED pour appeler un fichier, le charger en mémoire, l'afficher sur l'écran, placer le pointeur en fin de texte et passer en mode insertion :

```
XSUB
ED$1
 #A
 #T
 #L
I
```

Lorsque la commande XSUB a été utilisée une fois, il n'est pas nécessaire de la recharger en mémoire à chaque utilisation, et ce jusqu'au prochain démarrage à froid.

4

LA COMMANDE PIP

La commande PIP permet le transfert de fichiers entre unités CP/M. Les fichiers transférés peuvent l'être sous un nom différent lorsqu'il s'agit d'unités acceptant les noms de fichiers. Les fichiers peuvent également être concaténés à l'aide de cette commande. Un certain nombre de paramètres permettent en outre d'agir sur les fichiers en cours de transfert.

SYNTAXE

PIP est une commande non résidente. Le fichier PIP.COM doit donc se trouver sur une disquette dans un des lecteurs pour pouvoir être utilisé. PIP eut se présenter avec deux syntaxes différentes :

PIP <ENTER>

PIP <ligne de commande> <ENTER>

Dans le premier cas, PIP est chargé en mémoire, son indicatif est affiché sur l'écran (*), et le programme attend une ligne de commande. Le retour à CP/M s'effectue en tapant une ligne de commande vide (c'est-à-dire la touche <ENTER> en début de ligne). Dans le second cas, PIP est chargé en mémoire et la ligne de commande est exécutée. Une fois l'exécution terminée, le contrôle est renvoyé à CP/M.

LES LIGNES DE COMMANDES

La syntaxe d'une ligne de commande PIP est la suivante :

<destination> = <source1>, <source2>,...,<source n>
<parametre (s)>

<destination> peut être un nom de fichier précédé ou non d'un nom d'unité. <source1>,...,<source n> sont des noms de fichiers précédés ou non de noms d'unités, ou des noms d'unités qui sont copiés de gauche à droite vers la destination. La longueur de la ligne de commande ne doit pas dépasser 255 caractères. S'il y a plusieurs sources, il doit s'agir de fichiers ou d'unités contenant des caractères ASCII. Le dernier caractère doit être ↑Z. Les noms de fichiers peuvent contenir les caractères génériques * et ?. Le fichier destination peut être un des fichiers source. Dans ce cas, les fichiers

source sont d'abord concaténés (mis bout à bout), puis l'ensemble est copié dans le fichier destination.

EXEMPLES

FICH1 = FICH2

Copie le contenu du fichier FICH2 se trouvant sur l'unité par défaut dans le fichier FICH1 se trouvant également dans l'unité par défaut. Si le fichier FICH1 existe déjà, il est d'abord effacé, sauf s'il possède l'attribut $R/0 (voir commande STAT, Chapitre 3). Dans ce cas, le message suivant est affiché :

```
DESTINATION IS R/0, DELETE (Y/N)?
```

Si vous répondez Y, le fichier destination est effacé et la copie est effectuée. Dans le cas contraire, le fichier destination n'est pas modifié et la copie s'interrompt avec le message :

```
**NOT DELETED**
```

S'il n'y a pas assez de place sur la disquette, le message d'erreur suivant est affiché :

```
DISK WRITE ERROR: =FICH2
```

FICH1 = FICH2,FICH3

Concatène les fichiers FICH2 et FICH3 se trouvant sur l'unité par défaut et copie le tout dans le fichier FICH1, sur l'unité par défaut également.

A:FICH1 = B:FICH2

Copie le fichier FICH2 se trouvant dans l'unité B sur le fichier FICH1, dans l'unité A.

FICH1 = A:FICH2,B:FICH3

Concatène le fichier FICH2 se trouvant dans l'unité A et le fichier FICH3 se trouvant dans l'unité B, et place le tout dans le fichier FICH1, dans l'unité par défaut.

FICH1=FICH1,B:FICH2

Ajoute le contenu du fichier FICH2 se trouvant dans l'unité B au fichier FICH1 se trouvant dans l'unité par défaut.

FORMES ABRÉGÉES

Il est possible d'utiliser dans certains cas des commandes sous forme abrégée :

B:=FICH1

Copie le fichier FICH1 se trouvant sur l'unité par défaut et place la copie sur l'unité B, avec le même nom. (L'unité par défaut ne doit pas être l'unité B.)

A:=B:FICH1

Copie le fichier FICH1 de l'unité B vers l'unité A, sous le même nom.

FICH1=B:

Équivaut à FICH1=B:FICH1.

A:FICH1=B:

Équivaut à A:FICH1=B:FICH1.

UTILISATION DES UNITÉS LOGIQUES ET PHYSIQUES

La commande PIP permet également de copier des fichiers vers (ou à partir d') une unité logique ou physique. L'affectation des unités physiques aux unités logiques est fixée par défaut par la commande SETUP et peut être modifiée par la commande STAT (voir ces deux commandes au Chapitre 3). Les unités logiques sont :

CON : Console (unité d'entrée/sortie).

RDR : Unité de lecture (entrée).

PUN : Unité d'écriture (sortie).

LST: Unité de listage (sortie).

Les unités physiques disponibles sur l'Amstrad sont :

Écran	CRT:	UP2:		
Clavier	CRT:	UR2:		
Interface parallèle	LPT:			
1ere Unité d'entrée/sortie	TTY:			
2de unité d'entrée/sortie	UC1:	UL1:	UP1:	UR1:

Pour avoir la liste des autres unités physiques gérées par CP/M, reportez-vous à la description de la commande STAT au Chapitre 3.

La commande PIP possède en outre ses propres unités logiques :

NUL : Envoie quarante caractères "NUL" (ASCII 0). Cette unité était utilisée avec les périphériques lents (perforateurs de bandes papier).

EOF : Envoie un caractère ↑Z (fin de fichier texte).

INP : Entrée interne au programme PIP. Les données sont obtenues par une instruction CALL 103H et sont placées à l'adresse 109H (le bit de parité doit être nul).

OUT : Sortie interne de PIP. Les données sont placées dans le registre C. PIP exécute un CALL 106H. Les adresses 109H à 1FFH sont disponibles pour implanter des sous-programmes adaptés à des besoins spéciaux.

PRN : Cette unité est analogue à l'unité LST: avec les exceptions suivantes : les tabulations sont étendues à huit caractères, les lignes sont numérotées et un saut de page est ajouté toutes les soixante lignes (un saut de page est également ajouté en tête).

Dans le cas d'une copie à partir d'un périphérique (unité logique

ou physique), la copie ne se termine qu'à la réception d'un caractère ↑Z (ASCII 26).

Lors de toute copie, un fichier temporaire est créé pour recevoir le résultat de la copie. Si la copie réussit (et seulement dans ce cas), le fichier est renommé avec le nom spécifié dans la ligne de commande. En cas d'échec de la copie, et si le fichier destination existait déjà, il ne sera pas détruit. Par contre, si le fichier destination existe au début de la copie, il est nécessaire qu'un espace supplémentaire soit disponible sur la disquette pour recevoir le fichier temporaire. Dans le cas contraire, l'ordinateur affiche un message d'erreur :

```
DISK WRITE ERROR: = <nom du fichier source>
```

Lors d'une concaténation, tous les caractères ↑Z sont supprimés, sauf celui qui termine le dernier fichier à concaténer.

Lorsqu'une copie est faite à partir de l'unité RDR:, il est possible de la terminer par un ↑Z tapé au clavier.

Il est possible de concaténer des informations venant de fichiers et d'unités physiques ou logiques. Dans l'exemple suivant :

```
*FICH1=CON:,B:FICH2,RDR:,EOF:
```

le programme crée un fichier temporaire sur l'unité par défaut et y place tous les caractères tapés au clavier, jusqu'à réception d'un ↑Z. Le contenu du fichier FICH2 se trouvant dans l'unité B est ensuite ajouté, puis tous les caractères venant de l'unité d'entrée RDR:. Un caractère "fin de fichier" est ensuite ajouté (↑Z). Si l'opération s'est déroulée sans erreur, le fichier temporaire est renommé FICH1 (si un fichier FICH1 existait dans l'unité par défaut, celui-ci est d'abord effacé).

LES PARAMÈTRES

Toute opération de copie peut être modifiée par un ou plusieurs paramètres. Les paramètres doivent être placés entre crochets carrés ([et]) immédiatement après l'unité ou le fichier sur lesquels ils doivent intervenir. Les paramètres sont parfois suivis d'une valeur numérique.

Liste des paramètres de PIP :

B

Ce paramètre affecte le mode de transfert des données. Si ce paramètre est présent, les données sont gardées en mémoire jusqu'à réception d'un caractère ASCII XOFF. Les données sont alors transférées sur le disque. La quantité de données qui peut être stockée dépend de la capacité mémoire. Si la capacité est dépassée, un message d'erreur est affiché.

Dn

Le paramètre D suivi d'une valeur numérique n entraîne l'effacement de tous les caractères dépassant la colonne n au cours de la copie. Il est ainsi possible d'envoyer un texte à une imprimante en coupant les lignes trop longues.

E

Si ce paramètre est présent, toutes les données copiées sont affichées en même temps sur l'écran.

F

Avec ce paramètre, tous les caractères "saut de page" *(Form Feed)* se trouvant dans les données à copier sont supprimés.

Gn

Le paramètre G permet de copier un fichier se trouvant dans la zone utilisateur n vers la zone active. La zone active ne doit pas être la zone n. Le fichier PIP.COM doit se trouver dans la zone active. Pour copier le programme PIP dans une zone utilisateur, voir la commande USER au Chapitre 3.

H

Le paramètre H entraîne une vérification des données. Il est utilisé lors d'une copie de fichier .HEX. PIP vérifie alors que les données sont au format hexadécimal Intel. Les caractères inutiles sont supprimés. La correction des données est possible à partir du clavier.

I

Ce paramètre peut être utilisé pour supprimer, lors de la copie d'un fichier .HEX, tous les enregistrements ":00". L'utilisation de ce paramètre entraîne automatiquement celle du paramètre H.

L

Transforme tous les caractères majuscules en minuscules.

N

Si ce paramètre est présent, toutes les lignes copiées seront numérotées en commençant à 1. Les numéros sont sur cinq chiffres ; ils sont précédés par des blancs et suivis de deux-points (:). Si le paramètre N est suivi du chiffre 2, les numéros sont précédés des zéros non significatifs et ne sont pas suivis de deux-points.

O

Ce paramètre sert à transférer des fichiers non ASCII (fichiers objets). Le caractère "fin de fichier" est alors ignoré. Ce paramètre n'est pas nécessaire lorsque la source est un fichier sur disquette, car PIP connaît la longueur du fichier d'après le répertoire.

Pn

Le paramètre P suivi d'une valeur numérique n sert à insérer des sauts de page *(Form Feed)* toutes les n lignes. Si n est absent ou si n=1, les sauts de page seront insérés toutes les soixante lignes. Si le paramètre F est utilisé en même temps que le paramètre P, les sauts de page se trouvant dans le fichier à copier sont d'abord supprimés et de nouveaux sauts de page sont insérés.

Q<chaîne>↑Z

Si le paramètre Q est utilisé, la copie cesse dès que la chaîne de caractères <chaîne> est rencontrée (la chaîne de caractères doit être terminée par un ↑Z). La chaîne de caractères est présente dans la copie. Il est important de noter que CP/M transforme toute ligne

de commande en majuscules avant de l'exécuter. Dans la forme :

```
PIP FICH1=FICH2[QChaîne↑Z]
```

la chaîne de caractères sera d'abord transformée en majuscules avant exécution. La commande sera donc équivalente de :

```
PIP FICH1=FICH2[QCHAINE↑Z]
```

Pour copier jusqu'à une chaîne de caractères en minuscules, il faut d'abord charger PIP en mémoire en tapant :

```
PIP <ENTER>
```

puis taper la ligne de commande après l'indicatif de PIP. En effet, PIP ne transforme pas sa ligne de commande en majuscules avant exécution.

R

Le paramètre R permet de copier les fichiers système (attribut $SYS). Sans ce paramètre, les fichiers système ne sont pas reconnus.

S<chaîne>↑Z

Avec le paramètre S, la copie ne commence qu'à partir de la chaîne de caractères <chaîne> (terminée par un ↑Z). La chaîne de caractères est présente dans la copie. La remarque concernant les majuscules et les minuscules avec le paramètre Q est également valable ici.

Tn

Étend tous les caractères tabulation (↑I - ASCII 9) à n colonnes.

U

Transforme toutes les minuscules en majuscules lors de la copie.

V

L'utilisation de ce paramètre entraîne une vérification des données après la copie. Ce paramètre ne peut être utilisé qu'avec des copies de fichiers.

W

La présence de ce paramètre permet d'écrire sur un fichier protégé (attribut $R/0) sans qu'une confirmation soit demandée.

Z

Si ce paramètre est présent, tous les caractères copiés voient leur bit de parité forcé à 0. Un des usages de ce paramètre est la transformation des fichiers générés par le programme de traitement de texte WordStar en fichiers ASCII.

EXEMPLES D'UTILISATION DES PARAMÈTRES

PIP A:=B:FICH2[V]

Copie le fichier FICH2 de l'unité B vers l'unité A et effectue une vérification des données.

PIP A:FICH1=RDR:[FZ]

Copie dans le fichier FICH1 sur l'unité A tous les caractères venant de l'unité RDR: (par défaut, l'interface série) en supprimant les sauts de page et en forçant le bit de parité à 0.

PIP A:FICH1=B:FICH2[SCHAINE1↑ZQCHAINE2↑ZUP50E]

Copie dans le fichier FICH1 sur l'unité A, à partir du fichier FICH2 se trouvant dans l'unité B, tout le texte compris entre les chaînes de caractères CHAINE1 et CHAINE2, en transformant les minuscules en majuscules et en plaçant un saut de page toutes les cinquante lignes. Tous les caractères copiés sont en même temps affichés sur l'écran.

PIP FICH1:=FICH1[G2]

Copie le fichier FICH1 de la zone utilisateur 2 dans la zone utilisateur active (qui ne doit pas être la zone 2).

5

L'ÉDITEUR DE TEXTE ED

Pour créer ou modifier un fichier texte, il est nécessaire d'utiliser un programme appelé éditeur de texte. Un éditeur de texte est fourni avec CP/M. Il s'agit de la commande non résidente ED. Son utilisation est assez complexe et assez peu confortable. Il est préférable, lorsque cela est possible, d'utiliser un des nombreux éditeurs de texte modernes qui peuvent être achetés séparément. Cependant, ED étant souvent le seul éditeur disponible, il peut être utile d'en connaître le maniement.

ED est un *éditeur ligne,* c'est-à-dire qu'il permet de travailler sur une ligne à la fois. Les lignes de texte peuvent être d'une longueur quelconque (la seule limite étant la capacité de la mémoire) et doivent être terminées par un retour chariot (obtenu en pressant la touche ENTER). La taille des fichiers sur lesquels on peut travailler n'est limitée que par la capacité de la disquette, car ED charge en mémoire à la demande la partie du fichier qui doit être modifiée.

L'appel de l'éditeur se fait comme pour toute commande non résidente en tapant son nom, suivi du nom du fichier sur lequel on veut travailler. Si le fichier n'existe pas, il est créé par l'appel de la commande. Dans le cas contraire, il est simplement ouvert. En fait, dans tous les cas, les modifications sont écrites dans un fichier temporaire portant le même nom que le fichier spécifié et ayant l'extension .$$$. Lorsque l'on quitte ED, le fichier portant le même nom que le fichier de travail avec l'extension .BAK (s'il existe) est effacé, le fichier d'origine est renommé avec l'extension .BAK et le fichier temporaire est renommé avec l'extension d'origine. Le fichier .BAK contient donc toujours l'avant-dernière version du fichier. Si l'on désire annuler les modifications effectuées, il suffit d'effacer le fichier normal et de renommer le fichier .BAK avec l'extension normale. Par exemple, si l'on a modifié le fichier DUMP.ASM, une copie de la version originale a été créée sur la disquette sous le nom DUMP.BAK. Pour récupérer la version originale, il suffit de taper :

```
A>ERA DUMP.ASM
A>REN DUMP.ASM=DUMP.BAK
```

Il est préférable de s'assurer auparavant, à l'aide de la commande DIR, que le fichier DUMP.BAK existe bien sur la disquette.

Attention : si l'on effectue deux modifications successives d'un fichier, la copie portant l'extension .BAK ne contiendra plus la version originale mais la version modifiée une fois (l'avant-dernière version).

Par ailleurs, si un problème survient pendant la modification d'un fichier, celui-ci ne sera pas détruit, puisque le travail se fait sur un fichier temporaire. Ce surcroît de sécurité comporte un inconvénient majeur : dans le pire des cas, il peut exister simultanément sur la disquette trois versions du fichier portant les extensions .$$$, .BAK et l'extension normale. La contenance des disquettes étant limitée à 170 K (environ 170 000 caractères) dont une partie est occupée par CP/M et par le programme ED.COM, la taille maximale des fichiers utilisables par ED ne dépasse pas 50 K. Il est rarement nécessaire de travailler sur des fichiers aussi longs et toujours préférable de les diviser en morceaux plus petits et plus faciles à manipuler, qui pourront ensuite être assemblés (concaténés) à l'aide de la commande PIP (voir cette commande au Chapitre 4). Il est cependant possible, si le besoin s'en fait sentir, de travailler sur des fichiers plus longs, à condition de disposer de deux unités de disquettes.

On utilise alors la syntaxe suivante (par exemple) :

ED A:FICH1.TXT B:

Les données sont lues dans le fichier FICH1.TXT se trouvant dans l'unité A (unité par défaut) et les modifications sont enregistrées dans le fichier temporaire FICH1.$$$ sur l'unité B. Lorsque les modifications sont terminées, le fichier FICH1.TXT de l'unité A est renommé FICH1.BAK et le fichier FICH1.$$$ de l'unité B est renommé FICH1.TXT. Si l'on désire modifier une seconde fois le fichier, il faut utiliser la syntaxe :

ED B:FICH1.TXT A:

Cette fois, les données sont lues dans le fichier FICH1.TXT se trouvant sur l'unité B et les modifications sont enregistrées dans le fichier temporaire FICH1.$$$ qui est créé sur l'unité A. Une fois les modifications terminées, le fichier FICH1.BAK de l'unité A est effacé, le fichier FICH1.TXT de l'unité B est renommé FICH1.BAK et le fichier FICH1.$$$ de l'unité A est renommé FICH1.TXT. S'il existait déjà un fichier de même nom sur la seconde unité de disquettes, le programme affiche le message suivant :

```
FILE EXISTS
```

et se termine, empêchant la destruction accidentelle de celui-ci. Il

faut alors l'effacer à l'aide de la commande ERA ou utiliser une autre disquette avant de relancer la commande ED.

LES COMMANDES DE MANIPULATION DE TEXTE

Un premier groupe de commandes de ED sert à manipuler des lignes de texte pour les faire passer du fichier d'origine en mémoire centrale et de la mémoire centrale au fichier temporaire. Lorsque la commande ED est appelée, le fichier fourni comme argument de la commande est ouvert mais n'est pas chargé en mémoire. Les modifications du texte se font en mémoire à partir d'un emplacement déterminé par un pointeur. Six commandes sont disponibles pour la manipulation du texte. Dans la description des commandes, n représente un nombre entier positif ; +/—n représente un nombre entier positif ou négatif.

nA

La commande A charge n lignes de texte en mémoire à partir de la ligne suivant la dernière ligne déjà chargée, dans les limites de la mémoire disponible. S'il ne reste pas suffisamment de mémoire, il faut utiliser d'abord la commande W pour transférer les lignes déjà modifiées dans le fichier temporaire. La commande A doit être utilisée avant toute autre après l'appel de la commande ED. Si n est supérieur au nombre de lignes restant dans le fichier, le programme charge en mémoire les lignes restantes (dans la limite de la mémoire disponible). Si n est omis, la valeur prise par défaut est 1.

Deux options peuvent être utilisées pour remplacer n. Un signe # entraîne le chargement en mémoire de la totalité du fichier. Si le fichier ne tient pas en mémoire, le programme charge autant de lignes que possible. La valeur 0 entraîne le chargement d'autant de lignes que peut en contenir la moitié de la mémoire disponible.

La commande A ne modifie pas la position du pointeur. Lors de la première utilisation de cette commande, le pointeur est positionné au début de la première ligne.

nW

La commande W permet de transférer un certain nombre de lignes de la mémoire vers le fichier temporaire, permettant ainsi de libérer de la place en mémoire pour charger les lignes suivantes. Si

le pointeur se trouvait sur l'une des lignes à transférer vers le fichier, il est placé sur la première ligne du texte restant en mémoire. Dans le cas contraire, sa position n'est pas modifiée. Si n est supérieur au nombre de lignes se trouvant en mémoire, toutes les lignes sont transférées dans le fichier temporaire. Si n est omis, la valeur prise par défaut est 1.

Deux options peuvent être utilisées pour remplacer n. Un signe # entraîne le transfert dans le fichier temporaire de la totalité des lignes se trouvant en mémoire. La valeur 0 entraîne le transfert d'autant de lignes qu'il est nécessaire pour libérer la moitié de la mémoire.

E

La commande E termine une session de travail avec ED. Toutes les lignes se trouvant en mémoire sont transférées dans le fichier temporaire. La copie de sécurité (portant l'extension .BAK) est effacée. Le fichier original est renommé avec l'extension .BAK. Le fichier temporaire est renommé avec le nom du fichier original. La commande ED se termine et le contrôle est repassé à CP/M.

H

La commande H produit les mêmes effets que la commande E. Cependant, le contrôle n'est pas repassé à CP/M. Au lieu de cela, le nouveau fichier est ouvert et une nouvelle session de travail commence. Cette commande permet, en cas de travail très long, de sauvegarder régulièrement les états successifs du fichier. Ainsi, en cas de problème (panne de courant par exemple), un minimum de travail sera perdu. Il est prudent d'utiliser cette commande environ tous les quarts d'heure. Cependant, si la commande H est utilisée plusieurs fois, la version originale du fichier ne sera plus disponible. Seule l'avant-dernière version sera conservée.

O

La commande O permet d'annuler toutes les modifications effectuées depuis le début du travail ou depuis la dernière utilisation de la commande H. En raison de son caractère dangereux (risque de perte involontaire du travail effectué), cette commande doit être

confirmée. Le programme affiche le message suivant :

O-(Y/N) ?

Si vous répondez Y, la commande est exécutée. Toute autre réponse entraîne son annulation. Lorsque la commande O est exécutée, la mémoire est vidée. Pour travailler de nouveau sur le fichier, il faut utiliser la commande A. Les lignes sont alors chargées à partir du début du fichier.

Q

La commande Q produit les mêmes effets que la commande O. De plus, la session de travail est terminée. Le fichier temporaire est effacé et le contrôle est repassé à CP/M. Cette commande nécessite la même confirmation que la commande O.

DÉPLACEMENT DU POINTEUR ET AFFICHAGE DU TEXTE

Toute intervention sur le texte se fait à partir de la position du pointeur. Plusieurs commandes permettent de déplacer le pointeur et d'afficher le texte à partir de celui-ci.

+/-B

La commande B permet de placer le pointeur au début (+B) ou à la fin (-B) du texte. B (sans signe) est équivalent de +B.

+/-nL

La commande L permet de déplacer le pointeur de n lignes vers l'avant (+n) ou vers l'arrière (-n). n (sans signe) est équivalent de +n. Si n est trop grand par rapport à la taille du texte, le pointeur est déplacé au maximum dans le sens spécifié. Si n est omis, la valeur par défaut est 1. Le signe # peut remplacer n et signifie 65535. Il peut être précédé d'un signe + ou -. Son effet est de placer le pointeur en début ou en fin de texte. Il est préférable d'utiliser la commande B. Si n = 0, le pointeur est ramené au début de la ligne courante (voir plus loin le déplacement du pointeur dans une ligne, commande C).

+/—nT

La commande T permet d'afficher le texte à partir du pointeur. n lignes sont affichées à partir du pointeur (+ n) ou jusqu'au pointeur (— n). n est équivalent de + n. Si n=0, la ligne courante est affichée jusqu'au pointeur. Si n=1, la ligne courante est affichée à partir du pointeur (voir plus loin le déplacement du pointeur à l'intérieur d'une ligne, commande C). La touche ESC peut être utilisée pour stopper l'affichage. La commande T peut être combinée avec les commandes n: et :n. Par exemple :

```
10::20T
```

affiche les lignes 10 à 20 si elles se trouvent en mémoire. Le pointeur se trouve placé au début de la première ligne affichée.

+/—n

Cette commande est équivalente de +/—LT. Elle déplace le pointeur de n lignes vers l'avant ou vers l'arrière et affiche la nouvelle ligne courante.

Note :

Plusieurs commandes peuvent être placées sur la même ligne, comme dans l'exemple précédent. Il est ainsi possible de placer le pointeur sur une ligne donnée x en tapant :

```
ByLT
```

où y = x—1.

+/—nP

La commande P est censée afficher n pages de 23 lignes à partir de la position du pointeur si n est positif, ou n pages de 23 lignes précédant le pointeur si n est négatif. Cette commande a un fonctionnement plutôt aléatoire dans la version de ED livrée avec l'Amstrad et son utilisation est à déconseiller.

+/—nC

La commande C déplace le curseur de n caractères dans la ligne

courante vers l'avant ou vers l'arrière. Chaque fin de ligne compte pour deux caractères (retour chariot et passage à la ligne, CR-LF). Si le pointeur dépasse la fin de la ligne, la ligne suivante devient la ligne courante.

n:

Cette commande permet de placer le pointeur sur une ligne donnée en tapant simplement le numéro de la ligne suivi du caractère :.

:n

Cette commande permet de placer le pointeur sur la ligne n si celle-ci se trouve avant la position actuelle du pointeur et d'afficher cette ligne. Si la ligne n se trouve après la position du pointeur, celui-ci est placé sur la ligne n+1 et cette ligne est affichée.

EFFACEMENT ET INSERTION DE TEXTE

A partir de la position du pointeur, il est possible d'insérer ou de détruire des caractères en utilisant une des trois commandes prévues à cet effet.

I

La commande I permet de passer en mode insertion. Tous les caractères tapés après la commande I sont insérés à la position du pointeur. Le pointeur est avancé d'une position à chaque caractère tapé et se trouve donc toujours à la fin du texte inséré. La frappe de la touche ENTER insère une nouvelle ligne, sauf dans le cas où elle suit directement la commande I. Dans ce cas, il faut taper la touche ENTER une seconde fois pour insérer une ligne. L'insertion se termine par la frappe de ↑Z. Lors de l'insertion de texte, il est possible d'effacer le dernier caractère frappé en tapant ↑H. La touche DEL peut également être utilisée, mais elle produit à l'écran un résultat difficilement lisible. (Le caractère est effacé en mémoire, mais il est reproduit sur l'écran précédé d'un caractère identique au curseur, mais gris. L'état de la ligne en mémoire peut être vérifié avant de frapper la touche ENTER en tapant ↑R.) Le caractère ↑X annule la ligne en cours de frappe et renvoie le pointeur en début de ligne.

+/—nD

La commande D permet d'effacer des caractères à partir de la position du pointeur. Si n est positif, n caractères sont détruits après le pointeur. Si n est négatif, n caractères sont détruits avant le pointeur. Cette commande permet également de mettre deux lignes bout à bout afin de n'en faire plus qu'une. Il suffit pour cela de placer le pointeur en fin de ligne et d'effacer deux caractères en tapant 2D. Le passage à la ligne est en effet constitué de deux caractères (CR et LF), bien que ceux-ci soient générés par la frappe d'une seule touche (ENTER).

+/—nK

La commande K est utilisée pour effacer une ou plusieurs lignes. Si n est positif, n lignes sont effacées à partir de la position du pointeur. Si n est négatif, n lignes sont effacées avant la position du pointeur. Si le pointeur est situé au milieu d'une ligne, la commande K efface la fin de la ligne à partir du pointeur, y compris les caractères de fin de ligne (CR-LF). La commande -K efface le début de la ligne jusqu'au pointeur. Si le pointeur se trouve en début de ligne, la commande K efface la ligne, y compris les caractères de passage à la ligne.

RECHERCHE ET SUBSTITUTION DE TEXTE

nF<chaine de caractères>↑Z

Grâce à cette commande, il est possible de trouver dans le texte une chaîne de caractères quelconque. Si une valeur numérique n (positive) précède la commande, la $n^{\text{ième}}$ occurrence de la chaîne de caractères est recherchée. Si la chaîne cherchée est trouvée, le pointeur est placé après le dernier caractère de la chaîne. Si la chaîne n'est pas trouvée dans les lignes de texte se trouvant en mémoire, le pointeur n'est pas déplacé. Si la chaîne à rechercher comporte un saut de ligne, celui-ci doit être remplacé dans la commande par ↑L. Le caractère ↑Z sert à délimiter la chaîne si la commande F doit être suivie par d'autres commandes. Dans le cas contraire, il suffit de taper la touche ENTER.

nS<chaîne1>↑Z<chaîne2>↑Z

La commande S permet de remplacer automatiquement n fois la chaîne de caractères <chaîne1> par la chaîne <chaîne2>. A la fin de la substitution, le pointeur se trouve placé après la dernière chaîne remplacée. Si aucune substitution n'a eu lieu, le pointeur n'est pas déplacé.

nN<chaîne de caractères>↑Z

Cette commande est identique à la commande F lorsque tout le fichier se trouve en mémoire, ou lorsque la recherche aboutit dans le texte se trouvant en mémoire. Dans le cas contraire, le début du texte est transféré dans le fichier temporaire jusqu'à ce que la moitié de la mémoire soit libérée et la suite du texte est chargée en mémoire. La recherche continue alors jusqu'à son aboutissement ou jusqu'à la fin du fichier.

nJ<chaîne1>↑Z<chaîne2>↑Z<chaîne3>

La commande J prend trois chaînes de caractères pour arguments. La chaîne <chaîne1> est tout d'abord recherchée. Si elle est trouvée, la chaîne <chaîne2> est insérée juste après <chaîne1> et tout le texte se trouvant entre <chaîne2> et <chaîne3> est effacé. Comme dans les commandes F, S et N, une fin de ligne peut se trouver dans les chaînes de caractères et doit être entrée sous la forme d'un caractère ↑L.

Dans les commandes F, S, N et J, la valeur maximale de n est 100.

INSERTION DE TEXTE A PARTIR DE LA DISQUETTE

R<nom de fichier>↑Z

Il est possible d'insérer, dans le texte se trouvant en mémoire et à la position du pointeur, du texte provenant d'un fichier se trouvant sur la disquette. Le fichier doit avoir l'extension .LIB. Le caractère ↑Z à la fin de la commande n'est nécessaire que si la ligne doit comporter d'autres commandes. Dans le cas contraire, il suffit de taper la touche ENTER. La commande R sans paramètre peut être utilisée pour charger en mémoire, à la position du pointeur, le fichier X$$$$$$$.LIB créé par la commande X (voir cette commande).

TRANSFERT DE TEXTE DANS UN FICHIER

nX

Cette commande copie n lignes de texte à partir du pointeur dans un fichier sur la disquette portant le nom X$$$$$$$.LIB. Ce fichier peut être chargé en mémoire à la position du pointeur en utilisant la commande R sans aucun paramètre. La commande X n'efface pas le texte se trouvant en mémoire. Chaque nouvelle commande X ajoute le texte à transférer à la suite du fichier X$$$$$$$.LIB. La commande R n'efface pas le contenu du fichier X$$$$$$$.LIB. La commande 0X peut être utilisée pour vider ce fichier. Les commandes X et R peuvent donc être utilisées conjointement pour déplacer un bloc de lignes en mémoire. Le fichier X$$$$$$$.LIB est détruit lorsque l'on quitte l'éditeur.

MACRO-COMMANDES

nM<suite de commandes>↑Z

La commande M permet de répéter n fois une suite de commandes quelconque. Si n=0 ou n=1, la suite de commandes est répétée jusqu'à l'apparition d'une condition d'erreur (par exemple la fin du texte en mémoire). Le caractère ↑Z est nécessaire uniquement si la ligne doit comporter d'autres commandes. Exemple :

```
BMSjaune↑Zrouge↑Z-TT<ENTER>
```

Cette commande place le pointeur au début du texte en mémoire puis parcourt le texte en remplaçant chaque occurrence du mot "jaune" par le mot "rouge". A chaque substitution, la ligne concernée est affichée. Notez que, le pointeur se trouvant à un endroit quelconque de la ligne, la commande −T affiche le début de la ligne jusqu'à la position du pointeur et la commande T affiche la fin de la ligne à partir de la position du pointeur. A la fin des opérations, le pointeur se trouve juste après le dernier mot inséré. On obtiendrait exactement le même résultat en utilisant la commande suivante :

```
BMFjaune↑Z-5C5DIrouge↑Z0LT
```

Ici, la commande B place le pointeur au début du texte. La commande F trouve la première occurrence du mot "jaune" et place le curseur juste derrière. La commande −5C place le pointeur au début du mot et la commande 5D l'efface. La commande Irouge↑Z insère le mot "rouge". La commande 0L place alors le pointeur en début de ligne et la commande T affiche la ligne. L'ensemble de ces commandes (sauf la commande B, qui est en dehors de la macro-commande) est répété jusqu'à l'apparition d'une condition d'erreur, c'est-à-dire jusqu'à la fin du texte se trouvant en mémoire. La seule différence avec la commande précédente est qu'à la fin des opérations le pointeur se trouvera au début de la dernière ligne contenant le mot cherché.

VÉRIFICATION DE L'ESPACE MÉMOIRE ET NUMÉROS DE LIGNES

V

La commande V rend active la numérotation des lignes. (La numérotation est active par défaut au démarrage de la commande ED.)

−V

La commande −V supprime la numérotation des lignes.

0V

La commande 0V permet d'obtenir des informations concernant l'occupation de la mémoire. Ces informations sont affichées sous la forme :

```
20579/25015
```

indiquant (par exemple) que la capacité mémoire disponible est de 25 015 octets (caractères) et que 20 579 octets sont inoccupés.

La commande U

La commande U permet de forcer le texte entré en majuscules. Cette commande n'agit pas sur le texte précédemment tapé. La

commande —U permet de revenir au mode de fonctionnement normal.

La commande Z

La commande nZ suspend l'exécution du programme pendant environ n/4 secondes. Cette commande peut être insérée dans une macro-commande comportant des affichages de texte afin d'en faciliter la lecture.

LES MESSAGES D'ERREUR

En cas d'erreur, ED affiche un message d'erreur du type :

```
BREAK "x" AT c
```

dans lequel c est la commande au cours de laquelle l'erreur s'est produite et x un des caractères suivants :

- ? Caractère inconnu.
- > Dépassement de la capacité mémoire (utilisez la commande W pour libérer la mémoire).
- O Ouverture d'un fichier .LIB impossible (commande R).
- # La commande ne peut être exécutée le nombre de fois demandé.

EXEMPLES D'UTILISATION

Nous allons tout d'abord illustrer les commandes de manipulation de texte et de déplacement du pointeur en utilisant le fichier DUMP.ASM présent sur la disquette système CP/M. Ce fichier contient le listing source en assembleur de la commande DUMP.COM. Pour lancer la commande ED avec ce fichier, tapez :

```
A>ED DUMP.ASM
```

L'écran affiche

```
:*
```

ED est chargé en mémoire avec le nom du fichier à modifier. Avant toute modification, il est nécessaire de charger un certain nombre de lignes en mémoire à l'aide de la commande A. Tapez :

```
10A
```

pour charger dix lignes en mémoire. L'écran affiche maintenant :

```
1:*
```

indiquant que le pointeur se trouve sur la ligne 1. Nous pouvons afficher les lignes en mémoire grâce à la commande :

```
10T
```

Les dix lignes se trouvant en mémoire sont affichées :

```
 1: ;            FILE DUMP PROGRAM, READS AN INPUT FILE AND PRINTS IN HEX
 2: ;
 3: ;            COPYRIGHT (C) 1975, 1976, 1977, 1978
 4: ;            DIGITAL RESEARCH
 5: ;            BOX 579, PACIFIC GROVE
 6: ;            CALIFORNIA, 93950
 7: ;
 8:       ORG     100H
 9: BDOS  EQU     0005H  ;DOS ENTRY POINT
10: CONS  EQU     1      ;READ CONSOLE
 1:*
```

La dernière ligne affichée indique que le pointeur se trouve sur la ligne 1. Pour faire avancer le pointeur de douze caractères dans la ligne, tapez :

```
12C
```

L'affichage ne change pas. La commande T permet de vérifier le déplacement du pointeur. Tapez :

```
T
```

L'écran affiche le texte se trouvant entre le pointeur et la fin de la ligne :

```
PROGRAM, READS AN INPUT FILE AND PRINTS IN HEX
```

La commande

```
-T
```

permet d'afficher le texte se trouvant entre le début de la ligne et le pointeur :

```
:;          FILE DUMP*
```

L'astérisque (*) indique la position du pointeur. Lorsque le pointeur se trouve à un endroit quelconque d'une ligne, il est possible d'afficher la ligne entière en tapant :

```
-TT
```

ED exécute d'abord la commande −T affichant le début de la ligne, puis la commande T affichant la fin de la ligne. Dans ce cas, la position du pointeur n'est pas indiquée. Notez qu'il est toujours possible de taper plusieurs commandes sur une seule ligne. Les commandes tapées seront exécutées successivement. Nous allons par exemple renvoyer le pointeur en début de texte et afficher cinq lignes. Tapez :

```
B10T
```

L'écran affiche :

```
1: ;          FILE DUMP PROGRAM, READS AN INPUT FILE AND PRINTS IN HEX
2: ;
3: ;          COPYRIGHT (C) 1975, 1976, 1977, 1978
4: ;          DIGITAL RESEARCH
5: ;          BOX 579, PACIFIC GROVE
1: *
```

Nous pouvons charger en mémoire la totalité du fichier DUMP.ASM en tapant la commande :

```
#A
```

Le fichier étant suffisamment court, toutes les lignes sont char-

gées en mémoire. Si le fichier avait été trop long, ED aurait chargé en mémoire autant de lignes que possible. Nous pouvons maintenant afficher la totalité du fichier à l'aide de la commande :

```
#T
```

Il est possible de stopper l'affichage momentanément en tapant ↑S. L'affichage peut être repris en tapant une touche quelconque. Si une touche autre que ↑S est tapée pendant l'affichage, celui-ci est interrompu définitivement et le message :

```
BREAK "#" AT T
```

est affiché.

Pour placer le pointeur sur une ligne donnée, deux méthodes sont possibles. Le déplacement peut être relatif. On utilise alors la commande nL, qui déplace le pointeur de n lignes. Le déplacement se fait vers l'avant si n est positif et vers l'arrière si n est négatif. Tapez B pour ramener le pointeur en début de texte, puis :

```
10L
```

L'écran affiche le message :

```
11:*
```

indiquant que le pointeur se trouve sur la ligne 11. Cette ligne peut être affichée en utilisant la commande T :

```
11:    TYPEF EQU    2        ;TYPE FUNCTION
11:*
```

Nous pouvons ramener le pointeur à la ligne 5 et afficher cette ligne en tapant :

```
- 6LT
```

ce qui produit l'affichage suivant :

```
5: ;             BOX 579, PACIFIC GROVE
5:*
```

On peut également effectuer un déplacement absolu en tapant le numéro de la ligne où l'on veut placer le pointeur, suivi de deux points (:). Plaçons par exemple le pointeur à la ligne 20 en tapant :

```
20:
```

et affichons cette ligne à l'aide de la commande T. On aurait pu exécuter les deux commandes sur une seule ligne en tapant :

```
20:T
```

Pour effacer des caractères, nous allons utiliser la commande D. Placez le pointeur sur le début du mot DUMP, dans la première ligne de texte, et affichez cette ligne en tapant :

```
B7CT
```

L'écran affiche :

```
DUMP PROGRAM, READS AN INPUT FILE AND PRINTS IN HEX
```

Nous allons effacer le mot DUMP en tapant la commande :

```
5D
```

et nous utiliserons la commande −TT pour afficher le résultat :

```
1: ;          FILE PROGRAM, READS AN INPUT FILE AND PRINTS IN HEX
1:*
```

On peut réinsérer le texte supprimé en utilisant la commande I. Tapez :

```
IDUMP↑Z−TT
```

Nous allons maintenant couper la première ligne de texte en deux. Placez le pointeur au début du mot READS en tapant :

```
B16CT
```

L'écran doit afficher :

```
READS AN INPUT FILE AND PRINTS IN HEX
```

Tapez maintenant la commande I, puis deux fois la touche ENTER et ↑Z. Affichez ensuite les cinq premières lignes de texte en tapant :

```
B5T
```

L'écran affiche :

```
1:  ;                FILE PROGRAM,
2:  READS AN INPUT FILE AND PRINTS IN HEX
3:  ;
4:  ;                COPYRIGHT (C) 1975, 1976, 1977, 1978
5:  ;                DIGITAL RESEARCH
1:  *
```

La première ligne a été coupée en deux à la position du pointeur. Nous allons maintenant regrouper les deux premières lignes en une seule. Placez d'abord le curseur à la fin de la première ligne en tapant :

```
B21C
```

Il faut maintenant effacer le saut de ligne. Celui-ci est composé de deux caractères (CR-LF). Il faut donc effacer deux caractères en tapant :

```
2D
```

On peut constater le résultat en tapant les deux commandes :

```
BT
```

qui ont pour effet d'afficher la première ligne du texte. Le même résultat aurait pu être obtenu en tapant toutes les commandes sur une seule ligne :

```
B21C2DBT
```

Nous allons enfin effacer les deux premières lignes en tapant :

```
B2KT
```

L'écran montre que la troisième ligne du fichier porte maintenant

le n° 1. Les anciennes lignes 1 et 2 ont été effacées.

Nous pourrions sauvegarder la version modifiée du texte en utilisant la commande E. Au lieu de cela, nous allons abandonner les modifications en tapant :

```
Q
```

Une confirmation de la commande est demandée :

```
Q-(Y/N) ?
```

La touche Y confirme la commande et le fichier est abandonné. Le contrôle est repassé à CP/M. Toute autre touche annule la commande.

Les commandes de ED ne sont pas toujours faciles à utiliser. Il ne faut pas hésiter à décomposer les commandes et à utiliser fréquemment la commande d'affichage T pour contrôler l'évolution du travail. Il est également prudent d'employer assez souvent la commande H, qui permet de sauvegarder les versions intermédiaires du travail effectué. Ainsi, en cas d'accident, seul le travail effectué depuis la dernière sauvegarde sera perdu.

6

L'ASSEMBLEUR ASM

L'assembleur ASM constitue, avec l'éditeur ED, le programme de mise au point DDT et le chargeur LOAD, un système de développement complet livré en standard avec CP/M. ASM comporte bien sûr quelques limitations par rapport aux assembleurs plus modernes disponibles dans le commerce, mais il présente l'avantage d'être simple à utiliser et de ne pas nécessiter d'investissement supplémentaire. Sa première limitation vient du fait que CP/M a été développé à l'origine pour des systèmes construits autour du 8080 de Intel. Il s'agit donc d'un assembleur 8080 qui ne permet pas de tirer le meilleur parti des possibilités du Z80 de l'Amstrad. Son second handicap est de ne pas autoriser l'utilisation de macro-instructions comme le permettent les macro-assembleurs MACRO-80 de Microsoft et MAC de Digital Research. Les possibilités de ASM sont cependant suffisantes pour de nombreuses applications.

ASM permet donc d'assembler un programme source écrit en mnémoniques 8080 à l'aide d'un éditeur de texte quelconque (ED par exemple). L'assembleur est appelé par la commande :

ASM <nom de fichier>

ou

ASM<nom de fichier>.<paramètres>

Le fichier doit contenir un programme source écrit en mnémoniques 8080. Son nom peut être quelconque mais doit comporter l'extension .ASM. L'extension ne doit pas être mentionnée dans la commande. L'assembleur produit deux fichiers (si aucune erreur n'est rencontrée) :

<nom de fichier>.HEX

Contient le code machine en hexadécimal.

<nom de fichier>.PRN

Contient le listing d'assemblage du programme comportant le code source, le code machine généré et l'indication des éventuelles erreurs rencontrées.

Les erreurs rencontrées pendant l'assemblage sont également affichées sur l'écran.

Les paramètres peuvent être au nombre de trois. Ils sont séparés

du nom du fichier par un point et sont constitués d'une lettre indiquant les unités de disquettes où doivent se trouver les fichiers .ASM, .HEX et .PRN. Ainsi, la commande :

```
ASM DUMP. ABB
```

signifie que le programme à assembler est DUMP.ASM et qu'il se trouve dans l'unité A. Les fichiers DUMP.HEX et DUMP.PRN doivent être placés dans l'unité B.

Le paramètre X peut être utilisé pour envoyer le fichier .PRN vers l'écran (et simultanément vers l'imprimante si l'on a tapé précédemment ↑P).

Le paramètre Z permet de supprimer la création d'un des fichiers .PRN ou .HEX. La commande :

```
ASM DUMP.AZZ
```

assemble le programme source DUMP.ASM se trouvant sur la disquette dans l'unité A, mais ne génère aucun fichier. Cette option permet un assemblage beaucoup plus rapide et peut être utilisée pour mettre au point le programme source jusqu'à ce qu'il ne contienne plus d'erreur.

FORMAT DU PROGRAMME SOURCE

Le programme source est composé de lignes formées d'une combinaison des champs suivants :

N° Étiquette Instruction Opérande ;Commentaire

Chaque ligne doit être terminée par un retour de chariot (CR) et par un saut de ligne (LF), ou par un point d'exclamation (!). Le point d'exclamation permet donc de placer plusieurs instructions sur une même ligne physique.

Les différents champs doivent être séparés par une combinaison en nombre quelconque des caractères espace et tabulation.

Le numéro de ligne, qui doit être décimal, est optionnel. La plupart des éditeurs de programmes (comme ED) placent les numéros de lignes automatiquement et le programmeur n'a donc pas à s'en soucier. Les numéros de lignes sont ignorés par l'assembleur.

Les étiquettes sont optionnelles et peuvent avoir jusqu'à seize caractères, tous significatifs, sauf le caractère $, qui peut être uti-

lisé comme séparateur pour améliorer la lisibilité, comme dans l'exemple suivant :

SOUS$PROGRAMME

Les étiquettes peuvent être suivies ou non du caractère deux-points (:). Aucune différence n'est faite entre les caractères majuscules et minuscules.

L'instruction peut être un mnémonique 8080 ou une directive (pseudo-instruction) de l'assembleur, dont la liste sera donnée plus loin.

L'opérande peut être une constante, une étiquette, ou toute combinaison de ces deux catégories réalisée à l'aide des opérateurs arithmétiques et logiques dont la liste sera donnée plus loin.

Le commentaire est constitué d'une chaîne de caractères quelconque qui sera ignorée par l'assembleur et qui doit commencer par un point-virgule (;). Le programme source doit être terminé par la pseudo-instruction END. Toutes les lignes se trouvant après la pseudo-instruction END sont ignorées par l'assembleur.

LES OPÉRANDES

Les opérandes sont des expressions formées de combinaisons de constantes et d'étiquettes réalisées à l'aide des opérateurs arithmétiques et logiques. Ces expressions sont évaluées par l'assembleur. Le résultat de leur évaluation doit être cohérent avec l'instruction qui les précède. Certaines instructions nécessitent un opérande à 8 bits, d'autres un opérande à 16 bits, d'autres enfin une chaîne de caractères.

Les étiquettes présentes dans les expressions sont remplacées par la valeur qui leur a été attribuée lors de l'évaluation d'une instruction SET ou EQU, ou par l'adresse à laquelle elles font référence.

Les constantes numériques sont des valeurs sur 16 bits qui doivent être cohérentes avec les instructions qu'elles concernent et qui sont suivies d'un caractère indiquant la base dans laquelle elles sont exprimées :

B	binaire	(base 2)
O	octal	(base 8)
Q	octal	(base 8)
D	décimal	(base 10)
H	hexadécimal	(base 16)

En l'absence d'indication, l'assembleur suppose que la base est 10. Toutes les constantes doivent commencer par un chiffre. Les constantes hexadécimales commençant par une lettre doivent donc être précédées d'un zéro.

Des caractères $ peuvent être insérés dans les constantes pour améliorer la lisibilité. Si le résultat d'une évaluation dépasse le nombre de bits correspondant à l'instruction concernée, un message d'erreur est affiché.

LES OPÉRATEURS ARITHMÉTIQUES ET LOGIQUES

Les opérateurs suivants peuvent être utilisés dans les expressions constituant les opérandes :

+	somme arithmétique ou opérateur unaire (+a)
−	différence arithmétique ou opérateur unaire (−a)
*	multiplication
/	division
MOD	reste de la division
NOT	négation logique
AND	ET logique
OR	OU logique
XOR	OU logique exclusif
SHL	décalage à gauche d'un bit avec apport d'un zéro à droite
SHR	décalage à droite d'un bit avec apport d'un zéro à gauche

Les opérateurs sont considérés selon l'ordre de priorité décroissante suivant :

* / MOD SHL SHR
− +
NOT
AND
OR XOR

En cas de priorité égale, les opérateurs s'appliquent de gauche à droite. On peut modifier l'ordre d'application des opérateurs en utilisant des parenthèses.

LES MOTS RÉSERVÉS

Un certain nombre de mots correspondant aux registres du microprocesseur 8080 sont réservés et sont évalués comme indiqué ci-après :

A	7
B	0
C	1
D	2
E	3
H	4
L	5
M	6
SP	6
PSW	6

(Notez que l'assembleur ne fait aucune différence entre majuscules et minuscules dans les expressions comme dans les instructions, les étiquettes ou les indications de bases numériques.)

On peut utiliser les mnémoniques 8080 dans les expressions, où leur code numérique les remplace. Dans le cas d'instructions dont le code dépend de l'opérande, les bits non déterminés sont remplacés par des zéros.

Le caractère $ peut être utilisé dans une expression : il est remplacé lors de l'assemblage par l'adresse de l'instruction en cours d'assemblage.

LES CHAÎNES DE CARACTÈRES

Les constantes chaînes de caractères sont constituées de toutes combinaisons de caractères ASCII affichables et doivent être incluses entre apostrophes ('). Leur longueur ne doit pas excéder 64 caractères. Si une chaîne de caractères doit contenir une apostrophe, celle-ci doit être doublée (''). Les chaînes de caractères sont traitées par l'assembleur comme une suite de valeurs numériques correspondant aux codes ASCII des caractères. Les chaînes de caractères sont le seul cas où l'assembleur distingue les majuscules des minuscules.

Outre les mnémoniques des instructions du microprocesseur 8080, l'assembleur reconnaît un certain nombre de directives, ou pseudo-instructions, définissant des opérations à effectuer au moment de l'assemblage du programme.

La directive ORG

La directive ORG permet d'indiquer à l'assembleur l'adresse à laquelle il doit commencer l'assemblage du programme. Sa syntaxe est :

<étiquette>   ORG   <expression>

Le résultat de l'évaluation de l'expression doit être une valeur numérique sur 16 bits. Le programme sera assemblé à partir de cette adresse. L'étiquette est optionnelle et se voit attribuer la valeur de l'expression.

Les programmes devant être utilisés comme des commandes non résidentes doivent commencer par la directive :

```
ORG    100H
```

La directive END

La directive END indique à l'assembleur la fin du programme à assembler. L'assembleur assemble le programme jusqu'à la fin du fichier ou jusqu'à la première directive END rencontrée. Toutes les lignes placées après la directive END sont ignorées par l'assembleur. La syntaxe de la directive END est :

<étiquette>   END   <expression>

L'étiquette est optionnelle. Si l'expression est absente, l'adresse de démarrage du programme est par défaut 0000. Si une expression est présente, sa valeur doit être une adresse sur 16 bits. Cette adresse sera l'adresse de démarrage du programme et sera placée par l'assembleur à la fin du fichier .HEX. Une syntaxe courante dans les programmes CP/M est :

```
DEBUT ORG      100H
        .
        .
        .
      programme
        .
        .
        .
          END     DEBUT
```

La directive EQU

La directive EQU permet d'attribuer à une étiquette une valeur résultant de l'évaluation d'une expression. Sa syntaxe est :

<étiquette>   EQU   <expression>

Une étiquette donnée ne peut se voir attribuer une valeur qu'une seule fois au cours du programme. Dans le cas contraire, un message d'erreur est affiché. L'étiquette peut ensuite être utilisée dans le programme comme partie d'un opérande.

La directive SET

La directive SET sert, comme la directive EQU, à attribuer une valeur à une étiquette. Sa syntaxe est :

<étiquette>   SET   <expression>

La différence est que cette valeur peut être changée un nombre quelconque de fois au cours du programme. Cette directive est le plus souvent utilisée pour affecter une valeur logique à une étiquette contrôlant un assemblage conditionnel (voir les directives IF et ENDIF).

Les directives IF et ENDIF

Les directives IF et ENDIF servent à encadrer un groupe d'instructions qui seront ou non assemblées selon la valeur de l'expression suivant la directive IF. Leur syntaxe est :

```
IF <expression>
.
.
.
instructions
.
.
.
ENDIF
```

Si la valeur de <expression> est différente de 0, les instructions comprises entre IF et ENDIF sont assemblées. Dans le cas contraire, elles sont ignorées.

L'expression est souvent constituée d'une étiquette à laquelle une valeur a été affectée à l'aide de la directive SET.

La directive DB

La directive DB permet d'initialiser une zone de stockage de données en simple précision. Sa syntaxe est :

```
<étiquette>    DB    <expr1>,<expr2>,...,<expr n>
```

Les résultats de l'évaluation des expressions doivent être des valeurs sur 8 bits ou des chaînes de caractères. Si des chaînes de caractères entrent dans une expression composée, leur longueur ne doit pas excéder deux caractères. Les chaînes de caractères sont stockées sous forme d'une suite de codes ASCII dont le bit de parité (bit 7) vaut 0. L'étiquette est optionnelle et se voit attribuer pour valeur l'adresse de début de la zone de stockage. Exemple :

```
CR      EQU     13
LF      EQU     10
                .
                .
MESG    DB      "Tapez une touche quelconque",CR,LF
```

La directive DW

La directive DW fonctionne de la même façon que la directive DB, mais elle initialise une zone de stockage avec des valeurs en double précision (16 bits). Sa syntaxe est :

<étiquette> DW <expr1>,<expr2>,...,<expr n>

Les chaînes de caractères sont autorisées, mais leur longueur ne doit pas excéder deux caractères. Notez que les données sont stockées selon le format utilisé par le microprocesseur 8080, c'est-à-dire que l'octet de poids faible est placé en premier.

La directive DS

La directive DS est utilisée pour déclarer une zone de données qui ne doit pas être initialisée. Sa syntaxe est :

<étiquette> DS <expression>

L'étiquette est optionnelle et se voit attribuer pour valeur l'adresse de début de la zone de stockage. Le résultat de l'évaluation de l'expression doit être une adresse sur 16 bits.

LES INSTRUCTIONS DU 8080

La plus grande partie des lignes d'un programme en assembleur contient des mnémoniques d'instructions du 8080 que suivent ou non un ou plusieurs opérandes. La liste détaillée des mnémoniques 8080 et de leurs opérandes est donnée en Annexe 5.

LES MESSAGES D'ERREUR

Lorsqu'une erreur est rencontrée par l'assembleur, celui-ci place un code d'erreur en début de ligne dans le listing d'assemblage. Le code d'erreur et la ligne concernée sont également affichés sur l'écran. Voici la signification de chaque code :

D Erreur de donnée. Le format d'une donnée ne correspond pas au format de la zone où elle doit être placée.

E Erreur dans une expression. Une expression est mal formée et ne peut pas être évaluée.

L Erreur dans une étiquette. Une étiquette a été utilisée de façon incorrecte ou a été définie deux fois.

N Caractéristique prévue dans une version future de l'assembleur, mais non disponible dans la version utilisée (exemple : la directive TITLE).

O Une expression est trop complexe (contient trop d'éléments) pour pouvoir être évaluée.

P Erreur entre les deux passes de l'assemblage. Une étiquette a vu sa valeur modifiée entre les deux passes.

R Erreur de registre. Le registre spécifié comme opérande est incompatible avec l'instruction.

V Erreur de valeur. Un opérande est formé de façon incorrecte.

Plusieurs erreurs peuvent se produire en dehors de l'assemblage proprement dit et génèrent les messages d'erreur suivants :

NO SOURCE FILE PRESENT

Le fichier spécifié dans la commande ASM n'a pas été trouvé sur la disquette.

NO DIRECTORY SPACE

Le répertoire de la disquette est plein.

SOURCE FILE NAME ERROR

Le nom du fichier à assembler est formé de façon incorrecte.

SOURCE FILE READ ERROR

Le fichier contenant le programme à assembler n'a pu être lu correctement.

OUTPUT FILE WRITE ERROR

Une erreur s'est produite au moment de l'écriture des fichiers .HEX et .PRN. (Vérifiez qu'il reste de la place sur la disquette.)

```
CANNOT CLOSE FILE
```

Les fichiers .HEX et .PRN n'ont pu être fermés correctement. (Vérifiez que la disquette n'est pas protégée.)

EXEMPLE D'UTILISATION

Nous allons, à titre d'exemple, assembler un petit programme envoyant le caractère ↑G à la console. Le caractère ↑G produit un bip sonore. Appelez l'éditeur ED en tapant :

```
A>ED DEMO.ASM
```

L'écran affiche :

```
NEW FILE
     :*
```

Tapez I suivi de la touche ENTER pour passer en mode insertion, puis entrez les lignes suivantes. Les numéros de lignes et le caractère (:) qui les suit ne doivent pas être tapés. Ils sont placés automatiquement par l'éditeur. Chaque ligne doit être terminée par la frappe de la touche ENTER.

```
;PROGRAMME DEMO
;
DEBUT   ORG  100H
;
CTRL$G  EQU  7          ;CODE ASCII DU CARACTERE↑G
BDOS    EQU  5          ;POINT D'ENTREE DU BDOS
AFFCAR  EQU  2          ;FONCTION AFFICHAGE D'UN CARACTERE
;
        MVI  C,AFFCAR
        MVI  E,CTRL$G
        CALL BDOS
        JMP  0
        END  DEBUT
```

Quand toutes les lignes sont tapées, tapez ↑Z pour sortir du mode insertion, puis E suivi de la touche ENTER pour quitter l'éditeur.

Nous allons maintenant assembler le programme DEMO. Tapez :

```
A>ASM DEMO
```

L'écran affiche le message suivant :

```
CP/M ASSEMBLER – VER 2.0
010A
000H USE FACTOR
END OF ASSEMBLY
```

indiquant que l'assemblage s'est terminé sans erreur et que la prochaine adresse disponible est 10AH.

Vérifiez que les fichiers .HEX et .PRN ont bien été créés sur la disquette en tapant :

```
A>DIR DEMO.*
```

L'écran affiche :

```
A: DEMO     ASM:DEMO     BAK:DEMO     PRN:DEMO     HEX
```

Le programme est maintenant traduit en code hexadécimal dans le fichier DEMO.HEX. Il nous reste à faire de ce programme une image mémoire exécutable. Nous utiliserons pour cela la commande LOAD. Tapez :

```
A>LOAD DEMO
```

L'écran affiche :

```
FIRST ADDRESS      0100
LAST ADDRESS       0109
BYTES READ         000A
RECORDS WRITTEN      01
```

Ce message nous informe que le programme a été chargé à l'adresse 100H, qu'il se termine à l'adresse 109H, qu'il est long de 0AH octets (de 100H à 109H, il y a effectivement 0AH octets), et que le fichier DEMO.COM occupe 1 enregistrement sur la disquette. Il faut bien comprendre que le programme n'est pas réellement chargé en mémoire, mais qu'il est transformé en une image mémoire qui sera chargée effectivement lors de l'appel de la nouvelle commande non résidente DEMO.COM.

Pour tester le fonctionnement du programme, tapez :

```
A>DEMO
```

Le programme est chargé en mémoire et l'ordinateur émet un bip sonore, puis CP/M reprend le contrôle des opérations.

L'intérêt de ce programme est bien sûr extrêmement limité. Il peut cependant être intéressant d'examiner le listing d'assemblage (DEMO.PRN) pour mieux comprendre ce qui se passe au moment de l'assemblage. Les lecteurs qui voudraient approfondir leurs connaissances de l'assembleur sous CP/M liront avec profit le livre de A. Miller *CP/M approfondi,* publié chez Sybex.

7

LE PROGRAMME DE MISE AU POINT DDT

DDT est un programme de test et de mise au point interactive des programmes. DDT comporte douze commandes permettant d'assembler des mnémoniques directement en mémoire, d'afficher le contenu d'une zone mémoire, de remplir une zone mémoire avec une valeur donnée, de charger un programme en mémoire et de l'exécuter en mode normal ou en mode "pas à pas", de désassembler une zone mémoire, de déplacer des données, enfin d'examiner des adresses mémoire ou les registres du microprocesseur et d'en modifier le contenu.

Le lancement de DDT se fait en tapant :

```
A>DDT
```

ou

```
A>DDT <nom de fichier>
```

Si un nom de fichier est fourni comme paramètre, il doit être de type .HEX ou .COM. Le fichier est alors chargé en mémoire pour être examiné ou modifié. DDT se charge en mémoire à la place du CCP (juste en dessous du BDOS). Si le nom du fichier à examiner n'est pas fourni au moment de l'appel de DDT, il faut alors initialiser le nom du fichier puis le charger en utilisant successivement les commandes :

```
I<nom de fichier>
R
```

Une fois le fichier chargé, DDT affiche le message suivant :

```
DDT VERS 2.2.
NEXT PC
xxxx yyyy
```

La valeur xxxx (toutes les valeurs sont en hexadécimal) indique l'adresse située immédiatement après la dernière adresse occupée par le fichier. La valeur yyyy indique l'adresse de début du fichier. Lorsque le programme chargé en mémoire a été testé et mis au point, il doit être sauvegardé sur la disquette. DDT ne comporte pas de commande de sauvegarde. Il faut donc quitter DDT en tapant ↑C ou en utilisant la commande G0 et sauvegarder le programme se trouvant en mémoire grâce à la commande SAVE (pour savoir com-

ment calculer le nombre de pages mémoire à sauvegarder, se reporter à la description de la commande SAVE, au Chapitre 3).

Si le programme à tester est de type .COM, le registre PC du microprocesseur est chargé avec la valeur 100H. Si le programme est de type .HEX, le PC est chargé avec la valeur contenue dans le dernier enregistrement du fichier (cette valeur est générée par l'assembleur lorsqu'il rencontre une instruction END <adresse de départ>).

Si le programme nécessite des paramètres, ceux-ci ne peuvent être transmis au moment de l'appel de DDT. Il faut donc initialiser la zone des paramètres à l'aide de la commande I avant de tester le programme. La zone des paramètres commence à l'adresse 5CH. Si l'on examine cette zone après l'appel de DDT à l'aide de la commande D, on y découvre le nom du programme à tester. Cela est tout à fait normal, puisque ce nom était le paramètre de l'appel de DDT. C'est ce nom qu'il faut remplacer par le paramètre du programme chargé en mémoire. On notera que les noms de fichiers sont étendus à huit caractères à l'aide de blancs et que le point séparant le nom de l'extension n'est pas utilisé. (Pour plus de renseignements sur ce sujet, voir le livre de A.R. Miller *CP/M approfondi* publié chez Sybex.)

LES COMMANDES DE DDT

La commande A

La commande A sert à assembler des mnémoniques 8080 directement en mémoire. Sa syntaxe est :

A<adresse>

Les mnémoniques seront assemblés à partir de l'adresse spécifiée. DDT affiche la prochaine adresse libre et attend l'entrée d'un mnémonique suivi de ses éventuels opérandes. La fin d'une ligne est signalée par la frappe de la touche ENTER. L'assemblage se termine lorsque la touche ENTER est frappée sur une ligne vide. La validité des codes assemblés peut alors être testée en utilisant la commande inverse L, qui permet de désassembler le contenu d'une zone mémoire (voir cette commande). En cas de test d'un programme particulièrement long, il est possible que celui-ci déborde sur la partie de DDT contenant les fonctions d'assemblage et de

désassemblage. Dans ce cas, l'utilisation de ces commandes produit un message d'erreur.

La commande D

Cette commande permet d'afficher le contenu d'une zone mémoire en hexadécimal et en ASCII. Sa syntaxe est :

D
D<adresse de départ>
D<adresse de départ>,<adresse de fin>

Dans le premier cas, l'affichage commence à l'adresse suivant immédiatement la dernière adresse affichée par la précédente commande D. Si la commande D n'a pas déjà été utilisée, l'affichage commence à l'adresse contenue dans le registre PC. L'affichage est réalisé selon le format suivant :

```
xxxx hh hh hh hh hh hh hh hh hh hh hh hh hh hh hh hh cccccccccccccccc
```

La valeur xxxx est l'adresse du premier octet affiché sur la ligne. Les valeurs hh sont les contenus des adresses successives commençant à xxxx. Les caractères c affichés à droite sont la représentation ASCII de ces valeurs. Lorsqu'une valeur ne correspond pas à un caractère ASCII affichable, celui-ci est remplacé par un point. Toutes les lignes affichées commencent à une adresse multiple de 16, sauf éventuellement la première, si une adresse de départ a été fournie et n'est pas un multiple de 16 (les adresses multiples de 16 sont terminées par un 0). Si aucune adresse de fin n'est fournie, les lignes sont affichées par groupes de 12. Il est donc possible d'examiner toute la mémoire en tapant simplement la commande D plusieurs fois de suite. Si une adresse de fin est utilisée, DDT affiche le contenu de la zone mémoire comprise entre les adresses de début et de fin. On peut stopper momentanément l'affichage en tapant ↑S et le relancer en tapant une touche quelconque. On peut stopper l'affichage définitivement en tapant une touche quelconque (excepté ↑S).

La commande F

La commande F permet de remplir une zone mémoire donnée avec une valeur. Sa syntaxe est :

F<adresse de début>,<adresse de fin>,<valeur>

La zone mémoire comprise entre l'adresse de début et l'adresse de fin est remplie avec la valeur fournie comme troisième paramètre.

La commande G

La commande G est utilisée pour lancer l'exécution d'un programme. Sa syntaxe est :

G
G<adresse de départ>
G<adresse de départ>,<point d'arrêt>
G<adresse de départ>,<point d'arrêt1>,<point d'arrêt2>
G,<point d'arrêt>
G,<point d'arrêt1>,<point d'arrêt2>

Lorsqu'une adresse de départ est fournie, l'exécution commence à cette adresse. Dans le cas contraire, l'exécution commence à l'adresse contenue dans le registre PC du microprocesseur (on peut vérifier le contenu du PC à l'aide de la commande X). Il est possible de spécifier un ou deux points d'arrêt. Dans ce cas, l'exécution s'arrête avant l'instruction contenue à cette adresse. Les points d'arrêt sont supprimés automatiquement lors de l'arrêt du programme. A ce moment, DDT affiche l'adresse à laquelle était placé le point d'arrêt qui a été rencontré. Les registres du microprocesseur peuvent alors être examinés à l'aide de la commande X.

La commande I

La commande I permet d'insérer un nom de fichier à l'adresse 5CH. Ce nom de fichier peut être utilisé comme paramètre par le programme en cours de test. DDT utilise lui-même cette adresse pour ses propres paramètres. Ainsi, après l'appel de DDT avec un nom de fichier pour paramètre, l'examen de la zone mémoire commençant en 5CH révèle la présence du nom de ce fichier. La syntaxe de cette commande est :

I<nom de fichier>

Si le nom de fichier comporte l'extension .HEX ou .COM, l'instruction R peut ensuite être utilisée pour charger le fichier en mémoire.

La commande L

La commande L permet de désassembler une partie de programme contenue en mémoire. Sa syntaxe est :

L
L<adresse de départ>
L<adresse de départ>,<adresse de fin>

La première forme désassemble douze lignes de code à partir de l'adresse suivant celle de la dernière instruction désassemblée par la commande L précédente. La deuxième forme désassemble douze lignes de code à partir de l'adresse spécifiée. La troisième forme désassemble le code compris entre l'adresse de début et l'adresse de fin. On peut interrompre l'affichage définitivement en tapant une touche quelconque ou momentanément en tapant ↑S. Il faut noter que DDT est incapable de désassembler les instructions spécifiques du microprocesseur Z80 et affiche dans ce cas un message d'erreur.

La commande M

La commande M sert à déplacer en mémoire un bloc de données. Sa syntaxe est :

M<adresse de début de bloc>,<adresse de fin de bloc>,
<adresse destination>

Le premier paramètre est l'adresse de début du bloc à déplacer. Le deuxième paramètre est l'adresse de fin du bloc à déplacer. Le troisième paramètre est l'adresse à laquelle le bloc doit être déplacé.

La commande R

La commande R permet de charger en mémoire le fichier dont le nom a été précédemment initialisé par la commande I dans la zone

commençant à l'adresse 5CH. Ce fichier doit être un programme de type .HEX ou .COM. La syntaxe de la commande R est :

```
R
R<base>
```

Le paramètre <base>, qui est optionnel, est une valeur qui sera ajoutée à toutes les adresses du programme. (Le chargement ne doit toutefois jamais affecter les adresses 0000H à 00FFH, qui contiennent les paramètres du système.) Dans le cas d'un fichier .HEX, les adresses de chargement sont lues dans chaque enregistrement du fichier. Les fichiers de type .HEX doivent contenir du code hexadécimal au format Intel (produit par exemple par l'assembleur ASM). Les fichiers .COM sont par contre toujours chargés à partir de l'adresse 0100H et doivent contenir du code machine au format binaire tel qu'il est produit par les commandes LOAD ou SAVE. Après le chargement du programme, DDT affiche le message :

```
NEXT PC
xxxx yyyy
```

La valeur xxxx indique l'adresse suivant immédiatement la dernière adresse occupée par le programme. La valeur yyyy est l'adresse contenue par le registre PC du microprocesseur et sert de valeur par défaut aux commandes D, G, L, T et U. Cette valeur est toujours 0100H dans le cas des fichiers .COM. Dans le cas des fichiers .HEX, elle est lue dans le dernier enregistrement du fichier où elle a été placée par l'assembleur. (L'assembleur prend pour cette valeur le résultat de l'évaluation de l'expression suivant la directive END.)

Notez que la commande :

```
A>DDT <nom de fichier>
```

est équivalente de la séquence de commandes suivante :

```
A>DDT
-I<nom de fichier>
-R
```

La commande S

La commande S permet d'examiner des adresses mémoire et d'en modifier le contenu. Sa syntaxe est :

S<adresse>

DDT affiche l'adresse fournie comme paramètre à la commande suivie de la valeur de son contenu (toutes les valeurs sont en hexadécimal). L'utilisateur peut alors changer le contenu de l'adresse affichée en tapant une valeur comprise entre 00H et FFH, suivie de la frappe de la touche ENTER. Si la touche ENTER est frappée seule, le contenu de l'adresse affichée est préservé. Dans les deux cas, DDT affiche l'adresse suivante. Pour mettre fin à l'examen de la mémoire, il suffit de taper un point (.) suivi de la touche ENTER.

La commande T

La commande T sert à "tracer" un programme, c'est-à-dire à l'exécuter instruction par instruction (on dit "pas à pas"). Sa syntaxe est :

T
T<nombre de pas>

Dans le premier cas, une seule instruction est exécutée *après* l'affichage du contenu des registres du microprocesseur (il faut toujours se souvenir que l'affichage reflète l'état du microprocesseur *avant* l'exécution de l'instruction).

L'adresse contenant la prochaine instruction est affichée en fin de ligne sous la forme *xxxx. Cette valeur devient la valeur par défaut utilisée par la commande L et la valeur contenue dans le registre double HL est utilisée par défaut par la commande D. (La description complète de l'affichage du contenu des registres est donnée avec l'étude de la commande X.)

Si un nombre de pas n est donné comme paramètre à la commande, n instructions sont exécutées avant l'arrêt du programme. On peut arrêter l'exécution du programme à tout moment en tapant une touche quelconque.

En mode trace, le programme s'exécute environ 500 fois plus lentement que la vitesse normale.

La commande U

La commande U est analogue à la commande T, à une exception près : lorsqu'un nombre de pas n est fourni comme paramètre, n instructions sont exécutées, mais l'affichage de l'état des registres du microprocesseur n'est pas fait pour chaque instruction.

La commande X

La commande X permet d'examiner l'état des registres du microprocesseur. Sa syntaxe est la suivante :

X
X<registre>

Le <registre> peut être un des suivants :

C	indicateur de retenue (Carry)
Z	indicateur de zéro (Zero)
M	indicateur de signe (Minus)
E	indicateur de parité (Even parity)
I	indicateur de retenue interne (Interdigit carry)
A	accumulateur
B	registre double BC
D	registre double DE
H	registre double HL
S	pointeur de pile
P	compteur de programme

Si aucun paramètre n'est fourni, l'ensemble des registres est affiché sous la forme :

```
CzZzMzEzIz A=yy B=xxxx D=xxxx H=xxxx S=xxxx P=xxxx <instruction> <opérandes> <adresse>
```

Les valeurs z peuvent être 0 ou 1 et représentent le contenu de chaque it du registre d'état.

La valeur yy représente le contenu de l'accumulateur (valeur sur 8 bits). Les valeurs xxxx représentent le contenu des registres doubles dont les noms sont donnés ci-dessus. Toutes les valeurs sont en hexadécimal.

Si un registre est fourni comme paramètre à la commande, seul le contenu de ce registre est affiché et l'utilisateur peut le modifier en tapant une nouvelle valeur ou le laisser inchangé en frappant simplement la touche ENTER.

ANNEXES

1
LE CODE ASCII

Le code ASCII est donné en ordre numérique avec les valeurs en décimal et en hexadécimal et les codes de contrôle correspondants. La première partie (code 0 à 127) est le code ASCII standard. La signification qui est donnée est propre à CP/M. la seconde partie comprend le jeu de caractères étendu de l'Amstrad.

JEU DE CARACTÈRES ASCII STANDARD

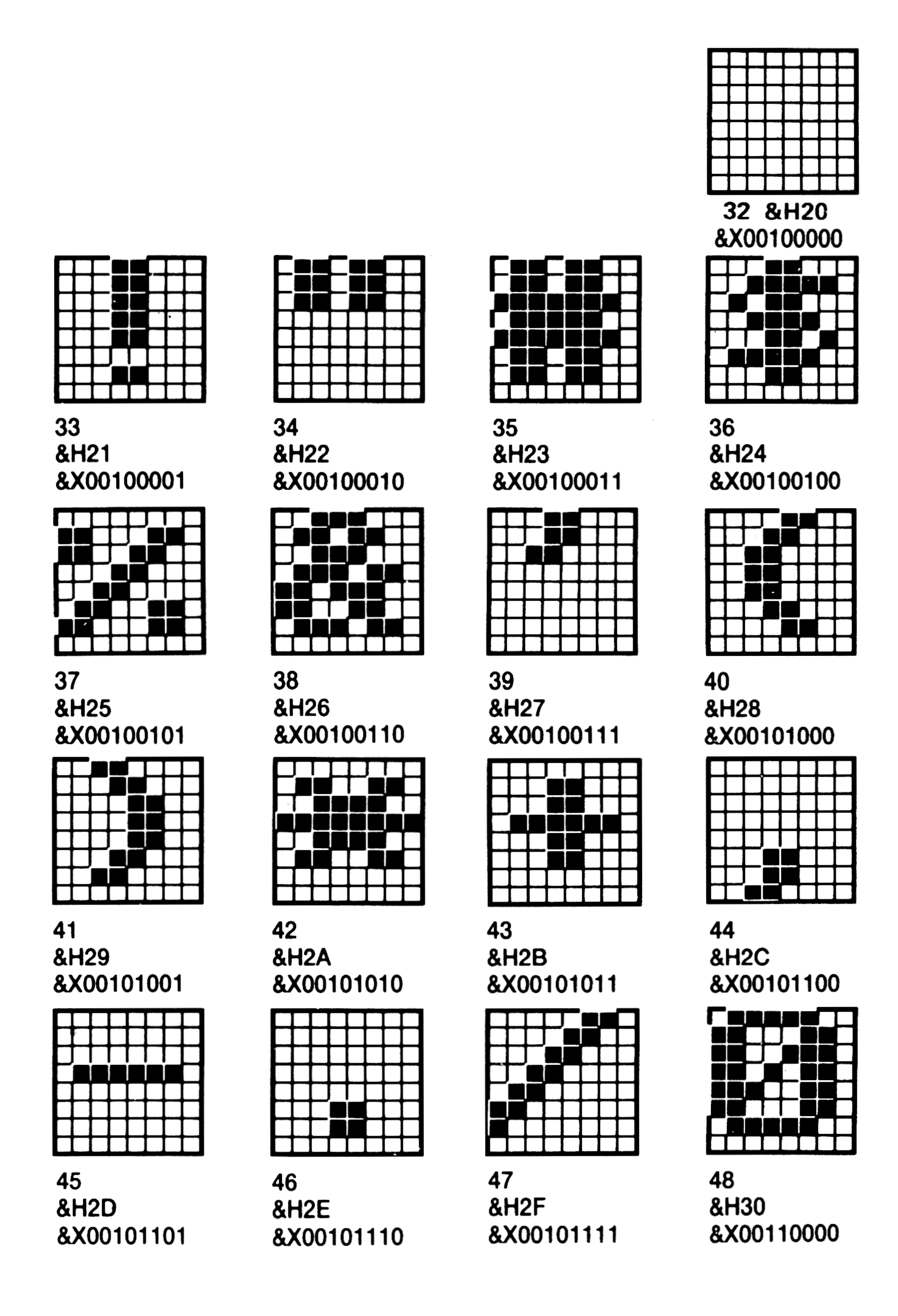
Caractère ASCII	Valeur décimale	Valeur hexadécimale	Caractère de contrôle	Signification
NUL	0	00	↑@	Caractère nul
SOH	1	01	↑A	
STX	2	02	↑B	
ETX	3	03	↑C	Démarrage à chaud
EOT	4	04	↑D	
ENQ	5	05	↑E	Fin de ligne physique
ACK	6	06	↑F	
BEL	7	07	↑G	Bip sonore
BS	8	08	↑H	Retour arrière
HT	9	09	↑I	Tabulation
LF	10	0A	↑J	Saut de ligne (fin de ligne logique)
VT	11	0B	↑K	
FF	12	0C	↑L	Saut de page
CR	13	0D	↑M	Retour chariot (fin de ligne logique)
SO	14	0E	↑N	
SI	15	0F	↑O	

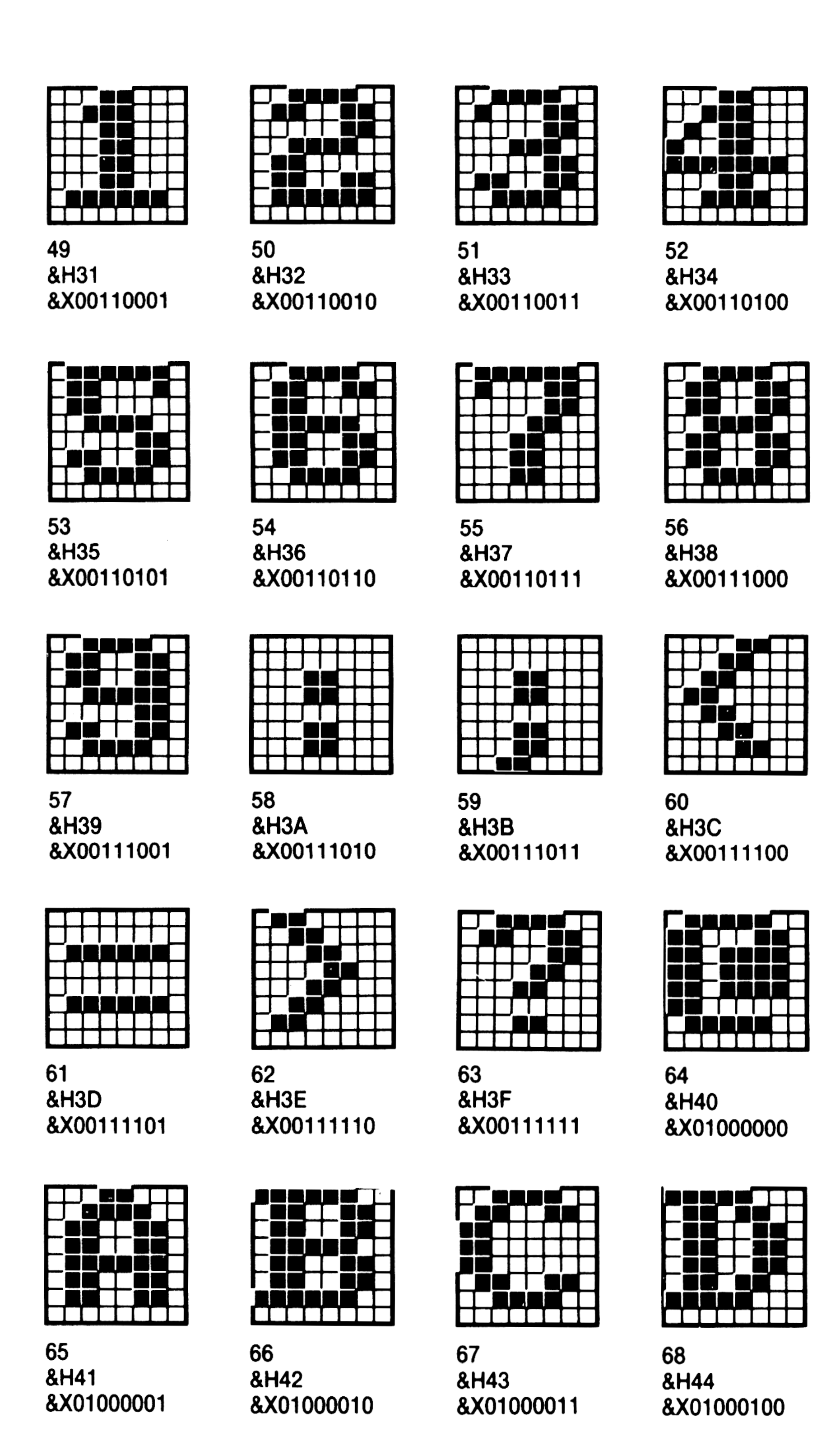
Caractère ASCII	Valeur décimale	Valeur hexadécimale	Caractère de contrôle	Signification
DLE	16	10	↑P	Début ou fin d'écho sur l'imprimante
DC1	17	11	↑Q	
DC2	18	12	↑R	Réaffichage de la ligne courante
DC3	19	13	↑S	Arrêt momentané de l'affichage
DC4	20	14	↑T	
NAK	21	15	↑U	Effacement de ligne
SYN	22	16	↑V	
ETB	23	17	↑W	
CAN	24	18	↑X	Effacement de ligne
EM	25	19	↑Y	
SUB	26	1A	↑Z	Fin de texte
ESC	27	1B	↑[	Escape
FS	28	1C	↑\	
GS	29	1D	↑]	
RS	30	1E	↑↑	
US	31	1F	↑_	
SP	32	20		Espace
!	33	21		
"	34	22		
#	35	23		
$	36	24		
%	37	25		
&	38	26		
'	39	27		Apostrophe
(	40	28		
)	41	29		
*	42	2A		
+	43	2B		
,	44	2C		Virgule
-	45	2D		Signe moins
.	46	2E		Point
/	47	2F		

Caractère ASCII	Valeur décimale	Valeur hexadécimale	Caractère de contrôle	Signification
0	48	30		
1	49	31		
2	50	32		
3	51	33		
4	52	34		
5	53	35		
6	54	36		
7	55	37		
8	56	38		
9	57	39		
:	58	3A		
;	59	3B		
<	60	3C		
=	61	3D		
>	62	3E		
?	63	3F		
@	64	40		
A	65	41		
B	66	42		
C	67	43		
D	68	44		
E	69	45		
F	70	46		
G	71	47		
H	72	48		
I	73	49		
J	74	4A		
K	75	4B		
L	76	4C		
M	77	4D		
N	78	4E		
O	79	4F		
P	80	50		
Q	81	51		
R	82	52		
S	83	53		
T	84	54		
U	85	55		
V	86	56		
W	87	57		

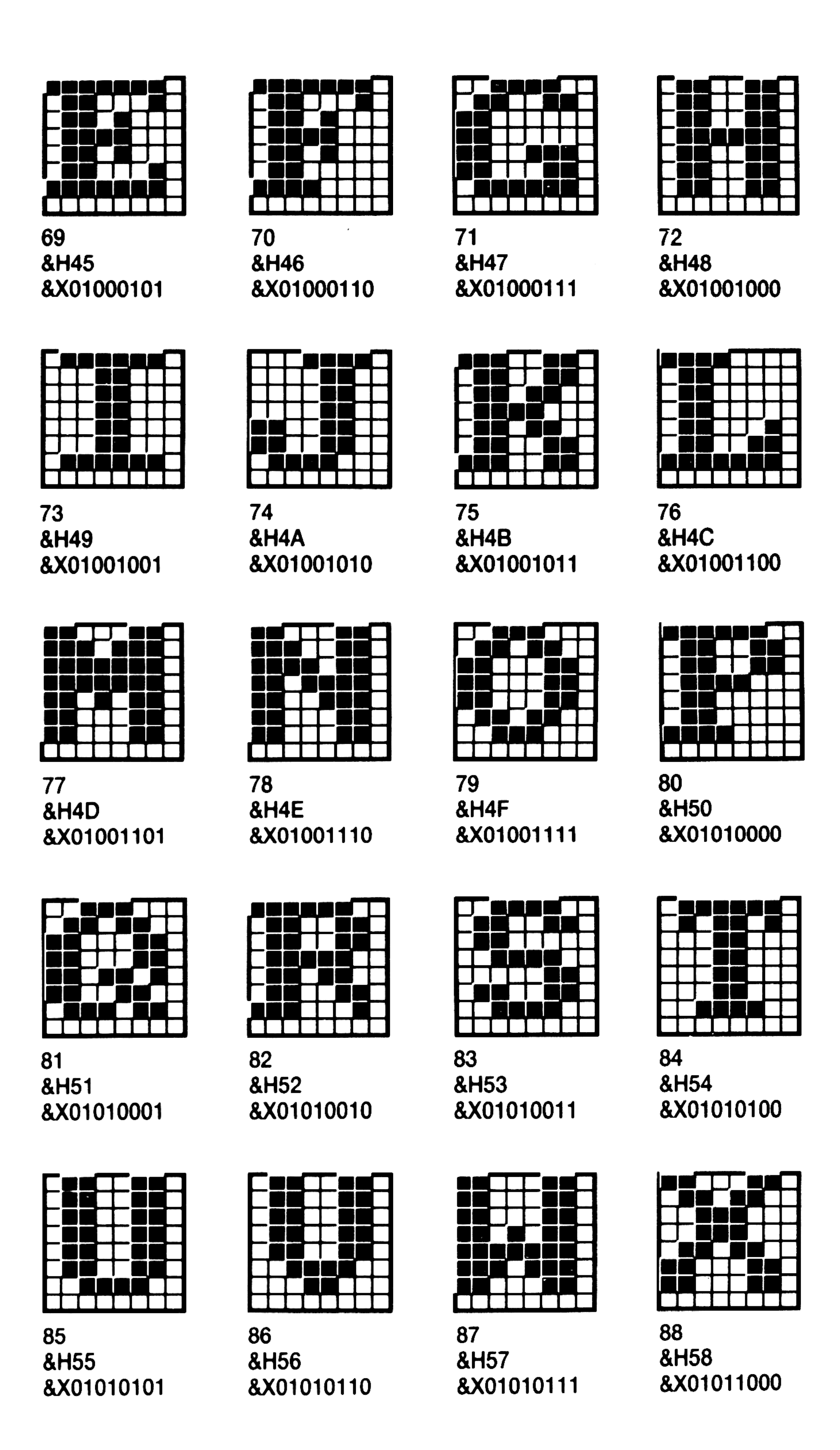
Caractère ASCII	Valeur décimale	Valeur hexadécimale	Caractère de contrôle	Signification
X	88	58		
Y	89	59		
Z	90	5A		
[	91	5B		
\	92	5C		
]	93	5D		
↑	94	5E		
_	95	5F		Souligné
`	96	60		Accent grave
a	97	61		
b	98	62		
c	99	63		
d	100	64		
e	101	65		
f	102	66		
g	103	67		
h	104	68		
i	105	69		
j	106	6A		
k	107	6B		
l	108	6C		
m	109	6D		
n	110	6E		
o	111	6F		
p	112	70		
q	113	71		
r	114	72		
s	115	73		
t	116	74		
u	117	75		
v	118	76		
w	119	77		
x	120	78		
y	121	79		
z	122	7A		
{	123	7B		
¦	124	7C		
}	125	7D		
~	126	7E		
DEL	127	7F		Effacement

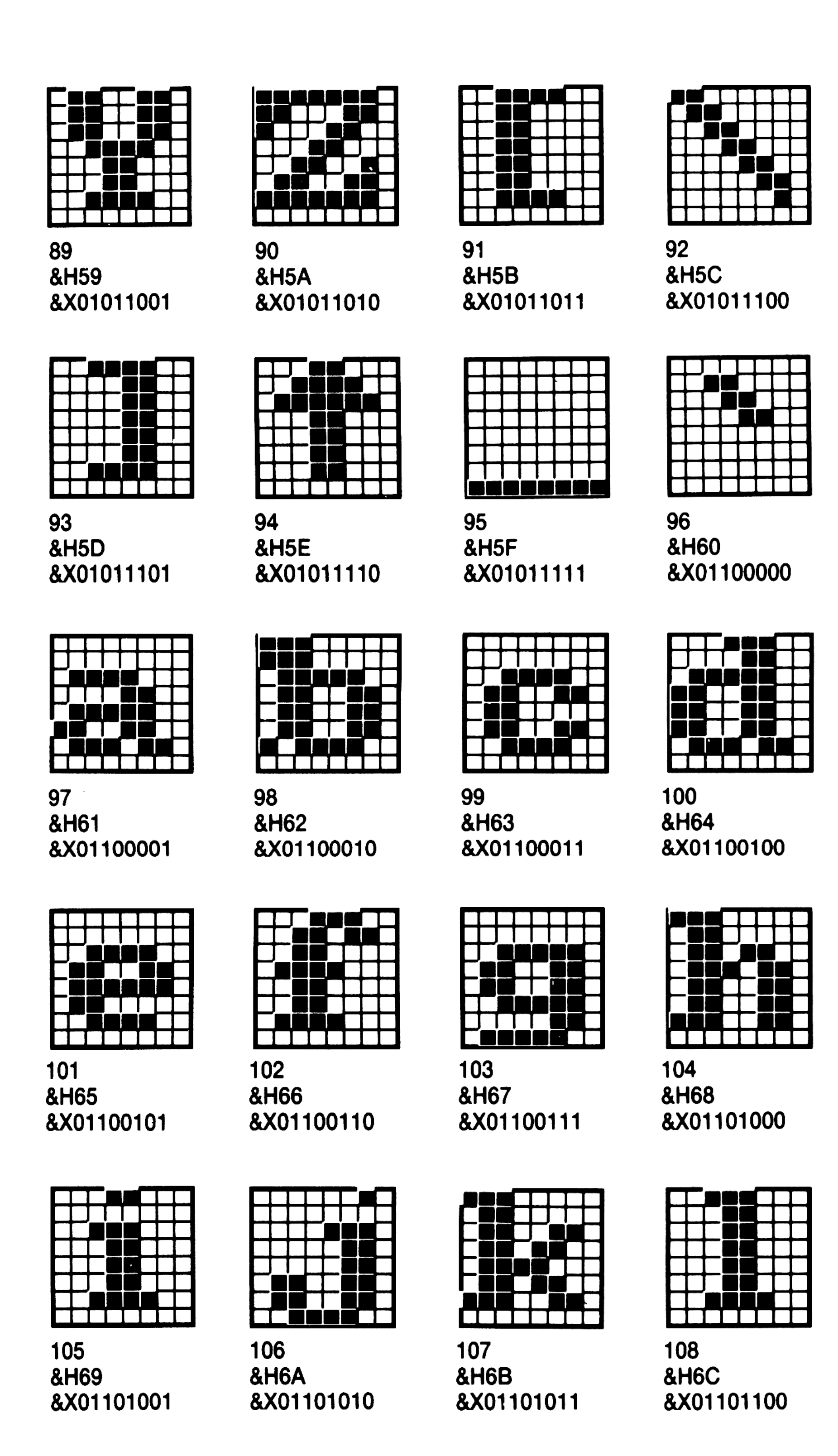
JEU DE CARACTÈRES ÉTENDU DE L'AMSTRAD



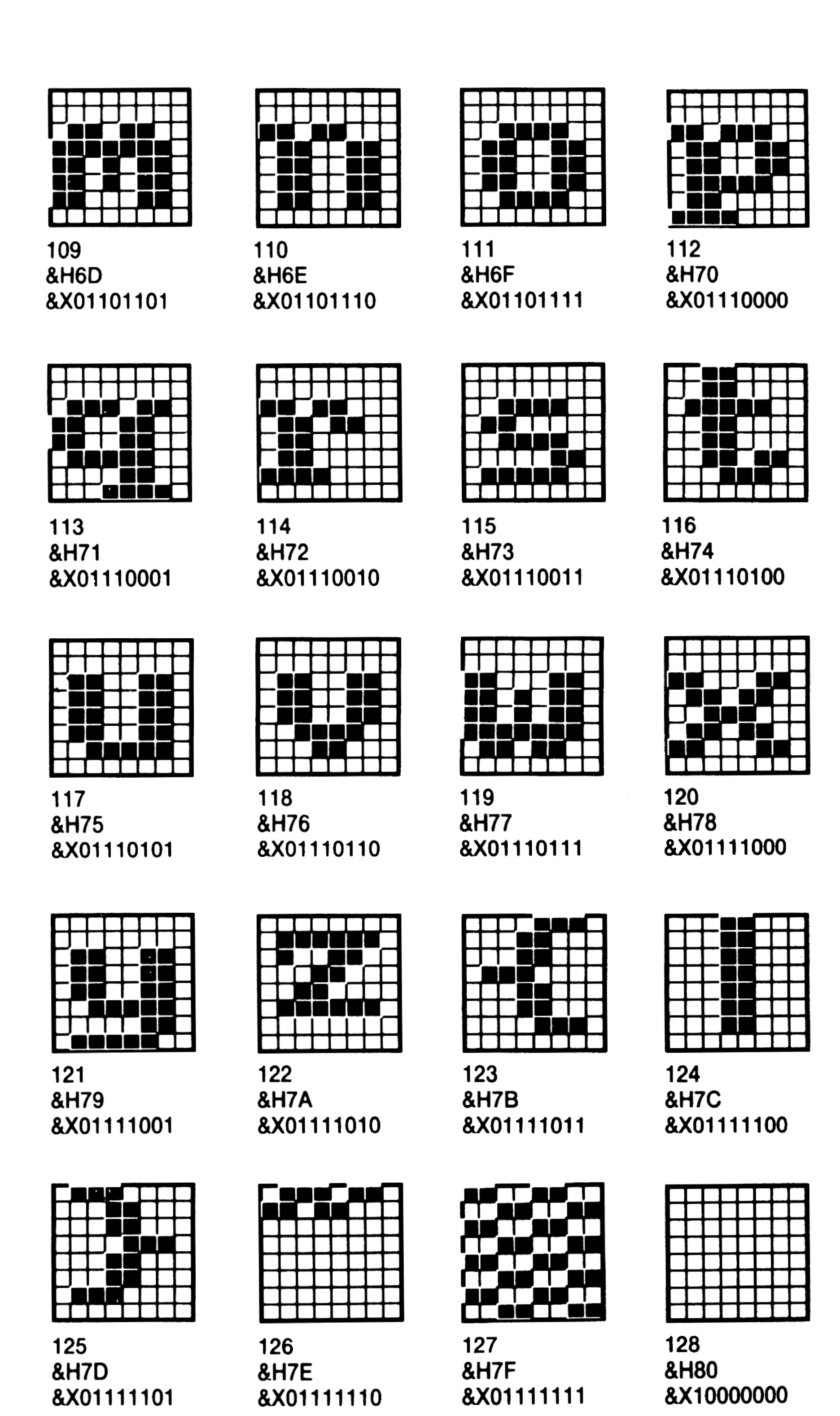


49
&H31
&X00110001
50
&H32
&X00110010
51
&H33
&X00110011
52
&H34
&X00110100
53
&H35
&X00110101
54
&H36
&X00110110
55
&H37
&X00110111
56
&H38
&X00111000
57
&H39
&X00111001
58
&H3A
&X00111010
59
&H3B
&X00111011
60
&H3C
&X00111100
61
&H3D
&X00111101
62
&H3E
&X00111110
63
&H3F
&X00111111
64
&H40
&X01000000
65
&H41
&X01000001
66
&H42
&X01000010
67
&H43
&X01000011
68
&H44
&X01000100

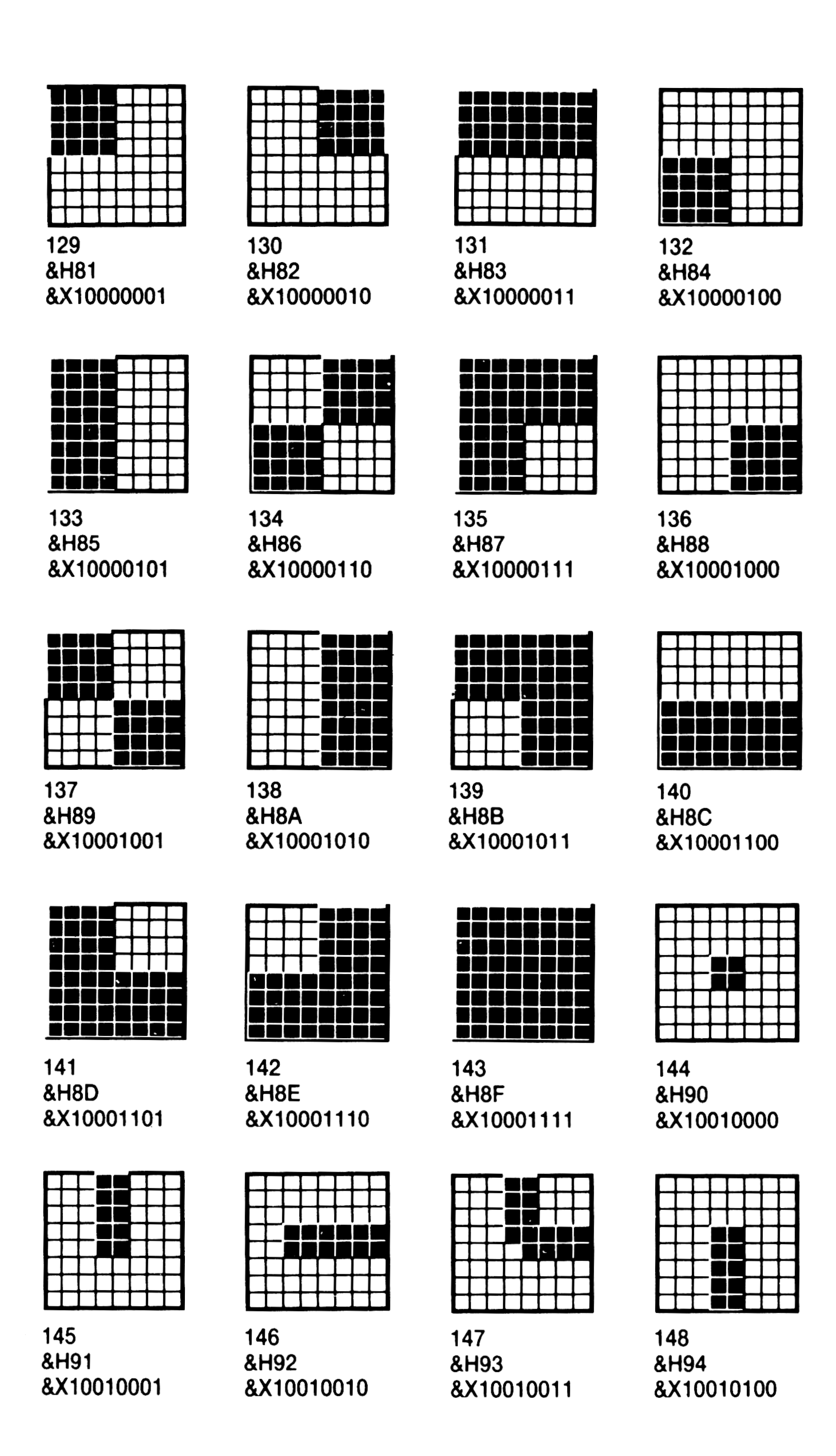

69
&H45
&X01000101
70
&H46
&X01000110
71
&H47
&X01000111
72
&H48
&X01001000
73
&H49
&X01001001
74
&H4A
&X01001010
75
&H4B
&X01001011
76
&H4C
&X01001100
77
&H4D
&X01001101
78
&H4E
&X01001110
79
&H4F
&X01001111
80
&H50
&X01010000
81
&H51
&X01010001
82
&H52
&X01010010
83
&H53
&X01010011
84
&H54
&X01010100
85
&H55
&X01010101
86
&H56
&X01010110
87
&H57
&X01010111
88
&H58
&X01011000

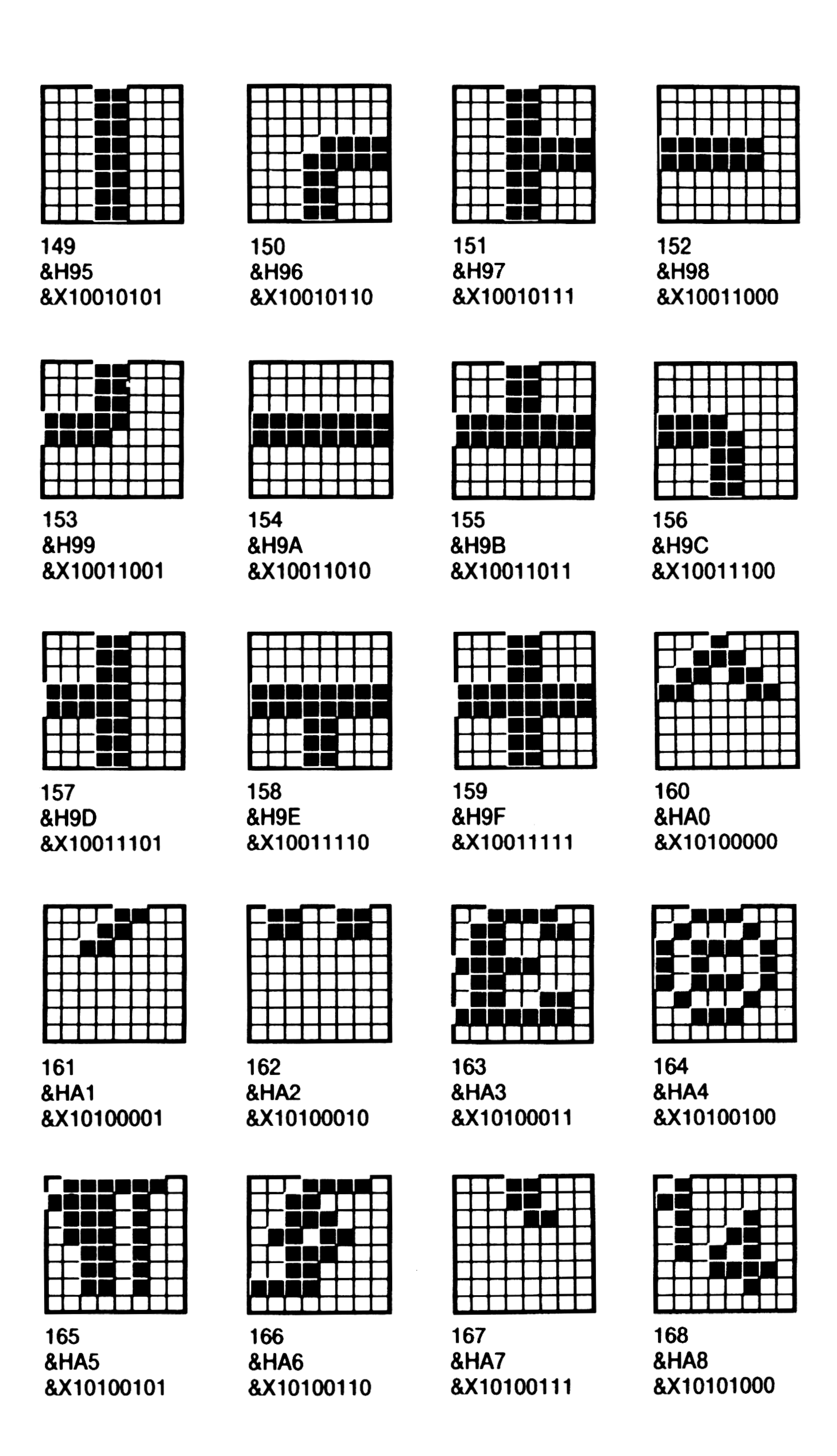


89
&H59
&X01011001
90
&H5A
&X01011010
91
&H5B
&X01011011
92
&H5C
&X01011100
93
&H5D
&X01011101
94
&H5E
&X01011110
95
&H5F
&X01011111
96
&H60
&X01100000
97
&H61
&X01100001
98
&H62
&X01100010
99
&H63
&X01100011
100
&H64
&X01100100
101
&H65
&X01100101
102
&H66
&X01100110
103
&H67
&X01100111
104
&H68
&X01101000
105
&H69
&X01101001
106
&H6A
&X01101010
107
&H6B
&X01101011
108
&H6C
&X01101100

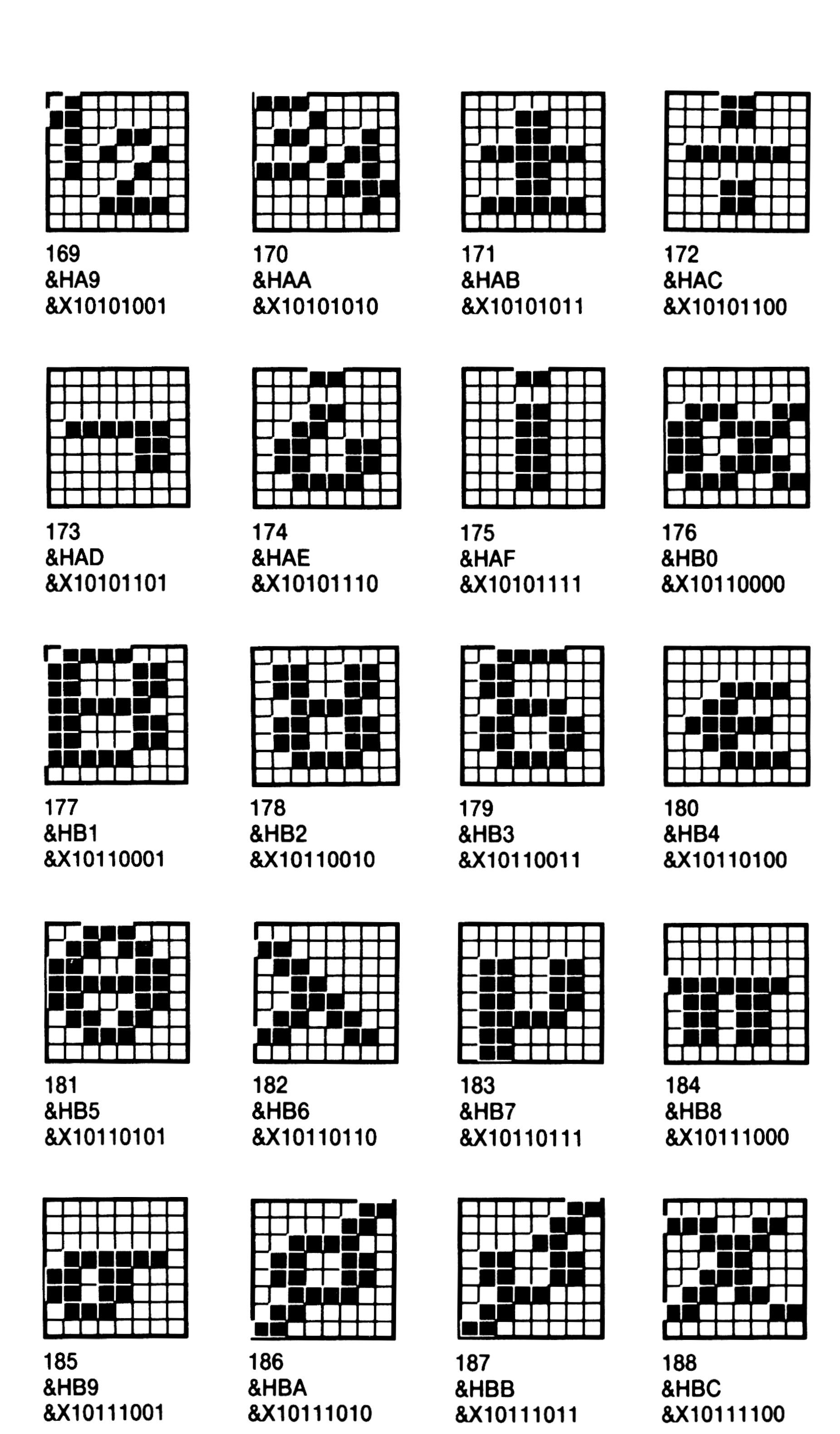


109
&H6D
&X01101101
110
&H6E
&X01101110
111
&H6F
&X01101111
112
&H70
&X01110000
113
&H71
&X01110001
114
&H72
&X01110010
115
&H73
&X01110011
116
&H74
&X01110100
117
&H75
&X01110101
118
&H76
&X01110110
119
&H77
&X01110111
120
&H78
&X01111000
121
&H79
&X01111001
122
&H7A
&X01111010
123
&H7B
&X01111011
124
&H7C
&X01111100
125
&H7D
&X01111101
126
&H7E
&X01111110
127
&H7F
&X01111111
128
&H80
&X10000000


129
&H81
&X10000001
130
&H82
&X10000010
131
&H83
&X10000011
132
&H84
&X10000100
133
&H85
&X10000101
134
&H86
&X10000110
135
&H87
&X10000111
136
&H88
&X10001000
137
&H89
&X10001001
138
&H8A
&X10001010
139
&H8B
&X10001011
140
&H8C
&X10001100
141
&H8D
&X10001101
142
&H8E
&X10001110
143
&H8F
&X10001111
144
&H90
&X10010000
145
&H91
&X10010001
146
&H92
&X10010010
147
&H93
&X10010011
148
&H94
&X10010100



149
&H95
&X10010101
150
&H96
&X10010110
151
&H97
&X10010111
152
&H98
&X10011000
153
&H99
&X10011001
154
&H9A
&X10011010
155
&H9B
&X10011011
156
&H9C
&X10011100
157
&H9D
&X10011101
158
&H9E
&X10011110
159
&H9F
&X10011111
160
&HA0
&X10100000
161
&HA1
&X10100001
162
&HA2
&X10100010
163
&HA3
&X10100011
164
&HA4
&X10100100
165
&HA5
&X10100101
166
&HA6
&X10100110
167
&HA7
&X10100111
168
&HA8
&X10101000



169
&HA9
&X10101001

170
&HAA
&X10101010

171
&HAB
&X10101011

172
&HAC
&X10101100

173
&HAD
&X10101101

174
&HAE
&X10101110

175
&HAF
&X10101111

176
&HB0
&X10110000

177
&HB1
&X10110001

178
&HB2
&X10110010

179
&HB3
&X10110011

180
&HB4
&X10110100

181
&HB5
&X10110101

182
&HB6
&X10110110

183
&HB7
&X10110111

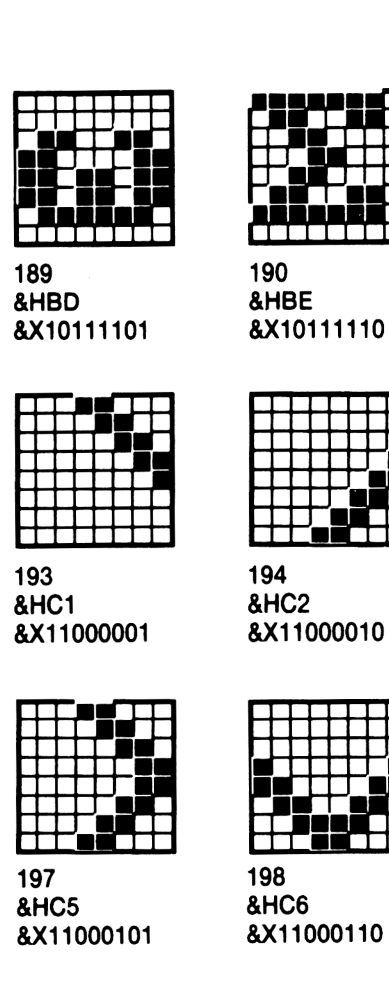
184
&HB8
&X10111000

185
&HB9
&X10111001

186
&HBA
&X10111010

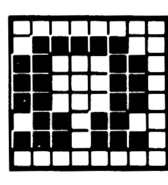
187
&HBB
&X10111011

188
&HBC
&X10111100

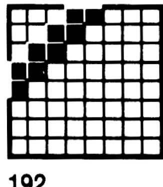


189
&HBD
&X10111101

190
&HBE
&X10111110



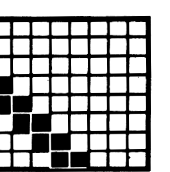
191
&HBF
&X10111111



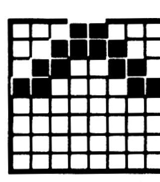
192
&HC0
&X11000000

193
&HC1
&X11000001

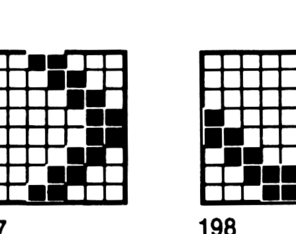
194
&HC2
&X11000010



195
&HC3
&X11000011

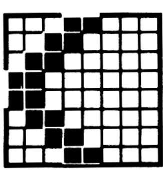


196
&HC4
&X11000100

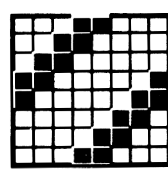


197
&HC5
&X11000101

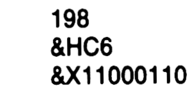
198
&HC6
&X11000110

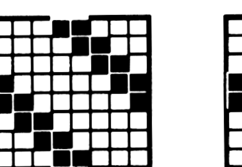


199
&HC7
&X11000111

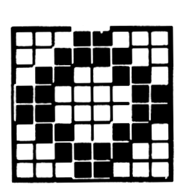


200
&HC8
&X11001000

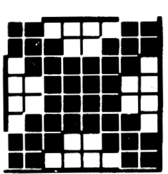




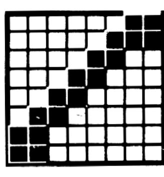
201
&HC9
&X11001001



202
&HCA
&X11001010

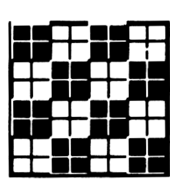


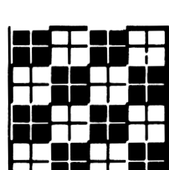
203
&HCB
&X11001011



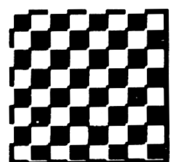
204
&HCC
&X11001100

205
&HCD
&X11001101

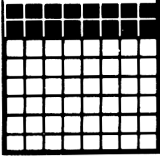




206
&HCE
&X11001110



207
&HCF
&X11001111



208
&HD0
&X11010000

209
&HD1
&X11010001

210
&HD2
&X11010010

211
&HD3
&X11010011

212
&HD4
&X11010100

213
&HD5
&X11010101

214
&HD6
&X11010110

215
&HD7
&X11010111

216
&HD8
&X11011000

217
&HD9
&X11011001

218
&HDA
&X11011010

219
&HDB
&X11011011

220
&HDC
&X11011100

221
&HDD
&X11011101

222
&HDE
&X11011110

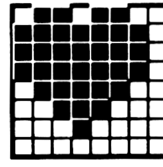
223
&HDF
&X11011111

224
&HE0
&X11100000

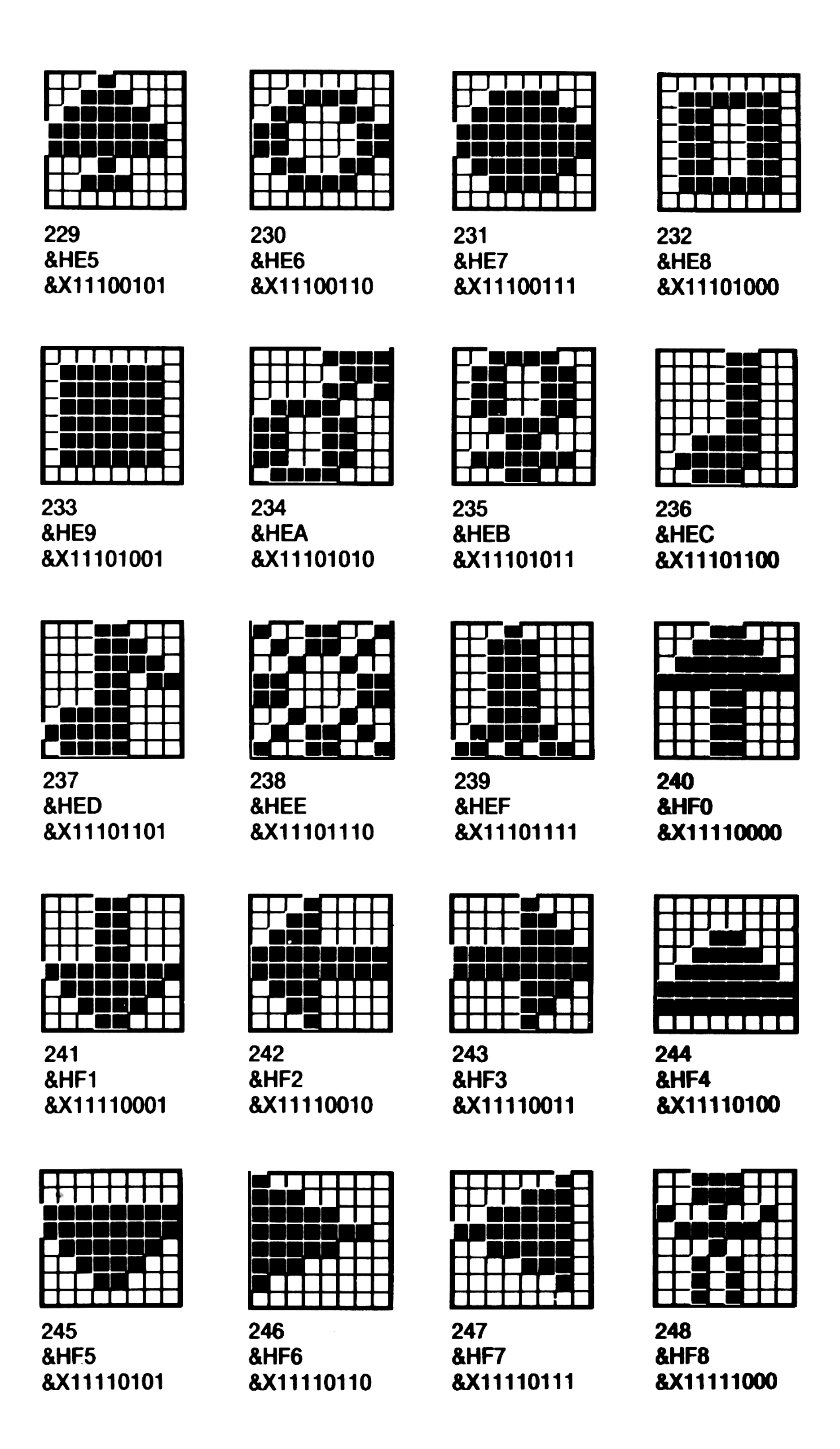
225
&HE1
&X11100001

226
&HE2
&X11100010

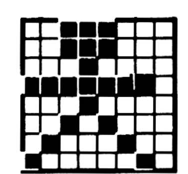
227
&HE3
&X11100011



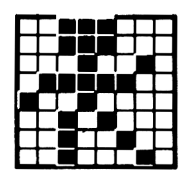
228
&HE4
&X11100100



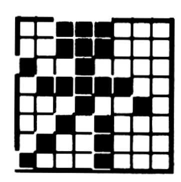
229
&HE5
&X11100101
230
&HE6
&X11100110
231
&HE7
&X11100111
232
&HE8
&X11101000
233
&HE9
&X11101001
234
&HEA
&X11101010
235
&HEB
&X11101011
236
&HEC
&X11101100
237
&HED
&X11101101
238
&HEE
&X11101110
239
&HEF
&X11101111
240
&HF0
&X11110000
241
&HF1
&X11110001
242
&HF2
&X11110010
243
&HF3
&X11110011
244
&HF4
&X11110100
245
&HF5
&X11110101
246
&HF6
&X11110110
247
&HF7
&X11110111
248
&HF8
&X11111000



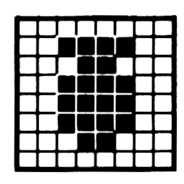
249
&HF9
&X11111001



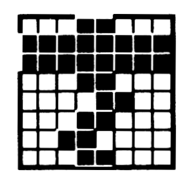
250
&HFA
&X11111010



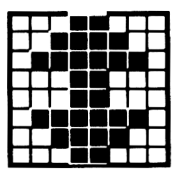
251
&HFB
&X11111011



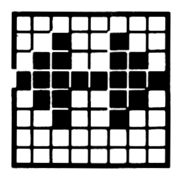
252
&HFC
&X11111100



253
&HFD
&X11111101



254
&HFE
&X11111110



255
&HFF
&X11111111

2
LA CARTE MÉMOIRE DE CP/M

L'implantation en mémoire de CP/M est différente suivant la quantité de mémoire qui lui est allouée par la commande MOVCPM. La table suivante donne l'implantation pour les configurations 20K, 24K, 32K et 48K. La configuration standard est 48K.

	20K	24K	32K	48K
CCP	3400H	4400H	6400H	A400H
BDOS	3C00H	4C00H	6C00H	AC00H
BIOS	4A00H	5A00H	7A00H	BA00H
Fin de la mémoire	4FFFH	5FFFH	7FFFH	BFFFH

La page 0 (adresses 0000H à 00FFH) est occupée par les paramètres du système selon l'organisation décrite ci-dessous :

0000H 0001H 0002H	Saut vers le point d'entrée BIOS
0003H	IOBYTE
0004H	Numéro d'unité active et de zone utilisateur active
0005H 0006H 0007H	Saut vers le point d'entrée du BDOS
0008H à 005DH	Zone de paramètres
005CH à 007FH	Zone de FCBs (descripteurs de fichiers)
0080H à 00FFH	Buffer à usage général

Les adresses 0000H à 0002H contiennent un saut au point d'entrée du BIOS. Un saut à l'adresse 0000H entraîne un démarrage à chaud de CP/M. L'adresse placée en 0001H et 0002H dépend de la taille de CP/M (voir la carte mémoire ci-dessus).

L'adresse 0003H contient l'IOBYTE. Chaque groupe de deux bits de l'octet placé à cette adresse détermine l'assignation d'une unité logique CP/M à une unité physique. Ces valeurs sont modifiées par la commande STAT.

L'adresse 0004H contient le numéro de l'unité de disquettes active et le numéro de la zone utilisateur active.

Les adresses 0005H à 0007H contiennent un saut au point d'entrée du BDOS. L'adresse contenue en 0006H et 0007H varie suivant la taille de CP/M. C'est l'adresse 0005H qui est utilisée pour tous les appels aux fonctions du BDOS. Le numéro de la fonction appelée doit se trouver dans le registre C. (Les fonctions du BDOS sont décrites à l'Annexe 4.)

La zone s'étendant de l'adresse 005CH à l'adresse 007FH contient les FCBs ou descripteurs de fichiers qui sont utilisés comme paramètres dans une commande CP/M (pour plus de détails, voir *CP/M approfondi*, de A.R. Miller, éditions Sybex).

La zone s'étendant de l'adresse 0080H à l'adresse 00FFH est un *buffer* qui peut contenir, entre autres, la fin de la ligne de commande.

3
ORGANISATION D'UNE DISQUETTE CP/M

L'Amstrad permet d'utiliser les disquettes selon trois formats différents : le format SYSTÈME, le format DONNÉES et le format IBM. Le format d'une disquette est automatiquement reconnu par CP/M au moment de la lecture ou de l'écriture.

CARACTÉRISTIQUES GÉNÉRALES D'UNE DISQUETTE

Chaque disquette comporte deux faces utilisables, mais une seule face peut être lue à la fois. Pour lire la seconde face, la disquette doit être retournée. (La tête de lecture travaille sur la face inférieure de la disquette.) Les disquettes sont donc considérées comme deux disquettes simple face indépendantes.

Chaque face comporte :

- 40 pistes numérotées de 0 à 39 ;
- un nombre de secteurs variable suivant les formats.

Chaque secteur comporte 512 octets.
L'unité CP/M est le bloc de 1 024 octets (2 secteurs).
Chaque disquette peut contenir un maximum de 64 fichiers.

LE FORMAT SYSTÈME

Dans le format SYSTÈME, chaque disquette comporte neuf secteurs numérotés de 41H à 49H. Les pistes 0 et 1 sont réservées au système. Une disquette doit être au format SYSTÈME pour pouvoir contenir CP/M.

Les pistes système sont organisées de la façon suivante :

piste 0 - secteur 41H	chargeur
piste 0 - secteur 42H	configuration
piste 0 - secteurs 43H à 47H	non utilisés
piste 0 - secteurs 48H à 49H	CCP et BDOS
piste 1 - secteurs 41H à 49H	CCP et BDOS

Le format VENDEUR est identique au format SYSTÈME, mais les pistes 0 et 1 sont inutilisées. La commande SYSGEN peut être utilisée pour transformer une disquette VENDEUR en disquette SYSTÈME.

LE FORMAT DONNÉES

Dans le format DONNÉES, chaque disquette comporte neuf secteurs numérotés de C1H à C9H. Aucune piste n'est réservée au système. CP/M ne peut pas être placé sur une disquette au format DONNÉES, ce qui limite son usage aux systèmes à deux disquettes.

LE FORMAT IBM

Une disquette au format IBM comporte huit secteurs numérotés de 1 à 8. Une piste est réservée. Ce format correspond logiquement au format des disquettes CP/M IBM.

4
LES FONCTIONS DU BDOS

L'accès à CP/M par un programme en assembleur se fait par l'intermédiaire de fonctions à l'aide d'un CALL à l'adresse 0005H. Le numéro de la fonction appelée doit être placé dans le registre C du microprocesseur. Les paramètres d'entrée et de retour sont fournis par l'intermédiaire de registres qui sont décrits ci-dessous.

Fonction 0 : Réinitialisation de CP/M

Condition d'entrée :
registre C = 00H

Cette fonction réinitialise CP/M et produit le même résultat qu'un saut à l'adresse 0000H.

Fonction 1 : Lecture d'un caractère au clavier

Condition d'entrée :
registre C = 01H
Condition de retour :
registre A = caractère ASCII tapé au clavier

Cette fonction lit un caractère au clavier. Le caractère lu est affiché sur l'écran. L'exécution du programme est suspendue tant qu'aucun caractère n'est tapé. CP/M prend en compte les caractères de contrôle.

Fonction 2 : Sortie d'un caractère sur l'écran

Conditions d'entrée :
registre C = 02H
registre E = caractère ASCII à afficher

Cette fonction affiche un caractère sur l'écran. CP/M prend en compte les caractères de contrôle qui auraient éventuellement été tapés au clavier.

Fonction 3 : Lecture d'un caractère sur l'unité logique RDR:

Condition d'entrée :
registre C = 03H
Condition de retour :
registre A = caractère ASCII entré

Cette fonction lit un caractère en entrée sur l'unité logique RDR:. Le programme est interrompu jusqu'à ce qu'un caractère soit lu.

Fonction 4 : Sortie d'un caractère sur l'unité logique PUN:

Conditions d'entrée :
registre C = 04H
registre E = caractère ASCII

Fonction 5 : Sortie d'un caractère sur l'unité logique LST:

Conditions d'entrée :
registre C = 05H
registre E = caractère ASCII

Fonction 6 : Entrée/sortie directe sur la console

Conditions d'entrée :
registre C = 06H
registre E = 0FFH (entrée)
caractère ASCII (sortie)
Condition de retour :
registre A = caractère ou état (entrée)

Cette fonction permet d'effectuer des entrées directes au clavier ou des sorties directes sur l'écran. Si le registre E contient la valeur 0FFH, il s'agit d'une entrée et le caractère entré est retourné dans le registre A. Si aucun caractère n'a été tapé au clavier, la valeur retournée est 00. Dans tous les cas, il n'y a pas d'interruption du programme. Si le registre E contient une valeur différente de 0FFH, le caractère correspondant est affiché à l'écran. Les caractères de contrôle ne sont pas pris en compte.

Fonction 7 : Lecture de l'IOBYTE

Condition d'entrée :
registre C = 07H
Condition de retour :
registre A = IOBYTE

Cette fonction permet de lire l'IOBYTE (adresse 0003H) qui contient l'affectation des unités physiques aux unités logiques.

Fonction 8 : Écriture de l'IOBYTE

Conditions d'entrée :
registre C = 08H
registre E = nouvelle valeur de l'IOBYTE

Cette fonction permet de changer l'affectation des unités physiques aux unités logiques en modifiant la valeur de l'IOBYTE.

Fonction 9 : Affichage d'une chaîne de caractères

Conditions d'entrée :
registre C = 09H
registre DE = adresse de la chaîne à afficher

Cette fonction permet d'afficher une chaîne de caractères stockée à partir de l'adresse indiquée par le registre DE. La chaîne de caractères doit être terminée par un caractère $.

Fonction 10 : Entrée console "bufferisée"

Conditions d'entrée :
registre C = 0AH
registre DE = adresse du buffer
Condition de retour :
Les caractères lus sont placés dans le buffer

Cette fonction permet la lecture d'une ligne sur la console en autorisant toutes les commandes d'édition de CP/M. Le premier caractère du buffer indique le nombre de caractères qui ont été lus

(maximum 255). La lecture se termine avec la frappe d'un saut de ligne (LF - ↑J) ou d'un retour chariot (CR - ↑M - touche ENTER) ou lorsque se produit un débordement du buffer.

Fonction 11 : Lecture de l'état de la console

Condition d'entrée :
registre C = 0BH
Condition de retour :
registre A = état de la console

Cette fonction retourne dans le registre A la valeur 0FFH si un caractère a été tapé et 00H dans le cas contraire.

Fonction 12 : Lecture du numéro de version de CP/M

Condition d'entrée :
registre C = 0CH
Condition de sortie :
registre HL = numéro de version

Fonction 13 : Réinitialisation du système de gestion de disquette

Condition d'entrée :
registre C = 0DH

Cette fonction permet de changer de disquette sans avoir à effectuer un démarrage à chaud.

Fonction 14 : Changement de l'unité de disquettes active

Conditions d'entrée :
registre C = 0EH
registre E = numéro de la nouvelle unité active

Les unités de disquettes sont numérotées 0 pour l'unité A et 1 pour l'unité B.

Fonction 15 : Ouverture d'un fichier

Conditions d'entrée :
registre C = 0FH
registre DE = adresse du FCB du fichier à ouvrir
Condition de sortie :
registre A = code répertoire

Le code répertoire indique que l'ouverture a été faite sans problème s'il est compris entre 0 et 3. S'il est égal à 0FFH, le fichier n'a pas été trouvé.

Fonction 16 : Fermeture d'un fichier

Conditions d'entrée :
registre C = 10H
registre DE = adresse du FCB du fichier
Condition de retour :
registre A = code répertoire du fichier

Si le code est compris entre 0 et 3, l'opération de fermeture a été effectuée avec succès. Dans le cas contraire, le code est 0FFH.

Fonction 17 : Recherche du premier nom de fichier

Conditions d'entrée :
registre C = 11H
registre DE = adresse du FCB du fichier à chercher
Condition de retour :
registre A = code répertoire

Cette fonction permet de trouver dans le répertoire d'une disquette le premier nom de fichier correspondant au FCB dont l'adresse se trouve dans le registre DE.

Fonction 18 : Recherche du nom de fichier suivant

Condition d'entrée :
registre C = 12H
Condition de retour :
registre A = code répertoire

Cette fonction permet de continuer la recherche commencée avec la fonction 17 dans la suite du répertoire.

Fonction 19 : Effacement d'un fichier

Conditions d'entrée :
- registre C = 13H
- registre DE = adresse du FCB du fichier à effacer

Condition de retour :
- registre A = code répertoire

Fonction 20 : Lecture séquentielle

Conditions d'entrée :
- registre C = 14H
- registre DE = adresse du FCB du fichier

Condition de retour :
- registre A = code répertoire

Cette fonction lit un enregistrement (128 caractères) du fichier dont le FCB se trouve à l'adresse référencée par le registre DE. Ces 128 caractères sont placés à l'adresse DMA courante. Cette adresse est fixée par défaut à 0080H à chaque démarrage de CP/M et peut être modifiée par la fonction 26. Le code répertoire est 00H si la lecture a été effectuée avec succès. Une valeur différente de 00H indique l'occurrence d'une erreur (fin de fichier par exemple).

Fonction 21 : Écriture séquentielle

Conditions d'entrée :
- registre C = 15H
- registre DE = adresse du FCB du fichier

Condition de retour :
- registre A = code répertoire

Cette fonction lit 128 caractères à l'adresse DMA et les place dans le fichier référencé par le FCB se trouvant à l'adresse contenue dans le registre DE.

Fonction 22 : Création d'un fichier

Conditions d'entrée :
registre C = 16H
registre DE = adresse du FCB du fichier à créer
Condition de retour :
registre A = code répertoire

Cette fonction crée une entrée dans le répertoire de la disquette correspondant au FCB construit préalablement à l'adresse référencée par le registre DE. Le code répertoire est 0, 1, 2 ou 3 si l'opération s'est déroulée correctement et 0FFH dans le cas contraire. L'ouverture du fichier n'est pas nécessaire après sa création.

Fonction 23 : Changement de nom d'un fichier

Conditions d'entrée :
registre C = 17H
registre DE = adresse du FCB du fichier
Condition de retour :
registre A = code répertoire

Les seize premiers octets du FCB contiennent l'ancien nom du fichier et les seize suivants contiennent le nouveau nom.

Fonction 24 : Lecture du vecteur "LOGIN"

Condition d'entrée :
registre C = 18H
Condition de retour :
registre HL = vecteur "LOGIN"

Chaque bit de la valeur retournée dans le registre HL indique si une des unités de disquettes a été ou non activée depuis le dernier démarrage de CP/M. (CP/M peut normalement adresser seize unités de disquettes.) Le bit 0 du registre L correspond à l'unité A et le bit 1 à l'unité B.

Fonction 25 : Recherche de l'unité de disquettes active

Condition d'entrée :
registre C = 19H
Condition de retour :
registre A = numéro de l'unité active

Le numéro est 0 pour l'unité A et 1 pour l'unité B.

Fonction 26 : Modification de l'adresse DMA

Conditions d'entrée :
registre C = 1AH
registre DE = adresse DMA

L'adresse DMA est l'adresse à laquelle sont placés les caractères lus dans un fichier. Sa valeur est fixée par défaut à 0080H à chaque démarrage de CP/M et à chaque réinitialisation du système de gestion de disquettes (fonction 13).

Fonction 27 : Lecture de l'adresse du vecteur d'allocation

Condition d'entrée :
registre C = 1BH
Condition de retour :
registre HL = adresse du vecteur d'allocation

Un vecteur d'allocation existe pour chaque disquette ayant été activée. Ce vecteur indique entre autres l'espace restant sur la disquette. La fonction 27 retourne dans le registre HL l'adresse de base du vecteur d'allocation correspondant à l'unité active.

Fonction 28 : Protection d'une disquette en écriture

Condition d'entrée :
registre C = 1CH

Fonction 29 : Lecture du statut des unités de disquettes

Condition d'entrée :
registre C = 1DH
Condition de retour :
registre HL = statut des unités de disquettes

Chaque bit de la valeur contenue dans HL indique si une unité de disquettes est protégée en écriture (1) ou non (0). Le bit 0 du registre L correspond à l'unité A et le bit 1 à l'unité B.

Fonction 30 : Modification des attributs de fichier

Conditions d'entrée :
registre C = 1EH
registre DE = adresse du FCB du fichier
Condition de sortie :
registre A = code répertoire

Fonction 31 : Lecture de l'adresse des paramètres disque

Condition d'entrée :
registre C = 1FH
Condition de retour :
registre HL = adresse

Fonction 32 : Lecture ou modification du numéro d'utilisateur

Conditions d'entrée :
registre C = 20H
registre E = 0FFH (lecture)
numéro d'utilisateur (modification)
Condition de retour :
registre A = numéro d'utilisateur (lecture)

Cette fonction permet de lire ou de changer le numéro de la zone utilisateur active. (Voir la commande USER.)

Fonction 33 : Lecture directe

Conditions d'entrée :
registre C = 21H
registre DE = adresse du FCB du fichier
Condition de retour :
registre A = code d'erreur

Cette fonction permet de lire un enregistrement dont le numéro est spécifié par les trois octets suivant le FCB (adresses FCB + 33, FCB + 34, FCB + 35, octet de poids faible en premier).

Fonction 34 : Écriture directe

Conditions d'entrée :
registre C = 22H
registre DE = adresse du FCB du fichier
Condition de retour :
registre A = code d'erreur

Cette fonction permet d'écrire directement un enregistrement donné référencé de la même façon que pour la fonction 33.

Fonction 35 : Calcul de la taille d'un fichier

Conditions d'entrée :
registre C = 23H
registre DE = adresse du FCB du fichier
Condition de retour :
La taille du fichier est placée dans les deux adresses suivant le FCB.

Cette fonction calcule la taille du fichier et l'inscrit dans les deux octets suivant le FCB. Le numéro d'enregistrement est fixé au numéro suivant le dernier enregistrement du fichier. Cette fonction peut être utilisée pour fixer le numéro d'enregistrement avant d'ajouter des informations dans un fichier en utilisant le mode séquentiel.

Fonction 36 : Calcul du numéro d'enregistrement en accès direct

Conditions d'entrée :
registre C = 24H
registre DE = adresse du FCB du fichier
Condition de retour :
numéro d'enregistrement fixé

Fonction 37 : Réinitialisation d'une unité de disquettes

Conditions d'entrée :
registre C = 25H
registre DE = vecteur de réinitialisation
Condition de retour :
registre A = 00H

Le vecteur de réinitialisation indique les unités à réinitialiser. Le bit 0 du registre E correspond à l'unité A et le bit 1 à l'unité B.

Fonction 40 : Écriture aléatoire avec zéros

Conditions d'entrée :
registre C = 28H
registre DE = adresse du FCB du fichier
Condition de retour :
registre A = code d'erreur

Cette fonction est analogue à la fonction 34, à la différence que si un bloc est écrit et n'était pas précédemment utilisé, il est préalablement rempli de zéros.

5

LES INSTRUCTIONS DU 8080

L'Amstrad utilise un microprocesseur Z80, mais l'assembleur ASM disponible sous CP/M est un assembleur 8080. Cette annexe décrit donc le jeu d'instructions du 8080. Les équivalents Z80 sont donnés entre crochets. Les noms des registres sont les noms standard du 8080, c'est-à-dire A, B, C, D, E, H, L et SP. Les registres doubles sont BC, DE et HL. Les indicateurs sont :

C retenue
H retenue interne (du bit 3 vers le bit 4)
N addition/soustraction
O débordement
P parité
S signe
Z zéro

r et r' désignent des registres 8 bits ;
nn désigne une constante à 8 bits ;
nnnn désigne une constante à 16 bits.

ACI nn <ADC A,nn>

Addition de la constante nn, de l'accumulateur et de l'indicateur de retenue C. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à zéro : N.

ADC M <ADC A, (HL)>

Addition du contenu de l'adresse référencée par le registre HL, de l'accumulateur et de l'indicateur de retenue C. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à zéro : N.

ADC r <ADC A,r>

Addition de l'accumulateur, du registre r et de l'indicateur de retenue C. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à zéro : N.

ADD M <ADD A, (HL)>

Addition du contenu de l'adresse référencée par le registre HL et de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à zéro : N.

ADD r <ADD A,r>

Addition de l'accumulateur et du registre r. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à zéro : N.

ADI nn <ADD A, nn>

Addition de la constante nn et de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à zéro : N.

ANA M <AND (HL)>

ET logique entre l'accumulateur et le contenu de l'adresse référencée par HL. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, N.
Indicateur mis à un : H.

ANA r <AND r>

ET logique entre l'accumulateur et le registre r. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, N.
Indicateur mis à un : H.

ANI nn <AND nn>

ET logique entre l'accumulateur et la constante nn. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, N.
Indicateur mis à un : H.

CALL nnnn <CALL nnnn>

Appel inconditionnel de sous-programme à l'adresse nnnn.
L'adresse de l'instruction suivante est placée sur la pile.

Indicateur modifié : aucun.

CC nnnn <CALL C,nnnn>
CM nnnn <CALL M,nnnn>
CNC nnnn <CALL NC,nnnn>
CNZ nnnn <CALL NZ,nnnn>
CP nnnn <CALL P,nnnn>
CPE nnnn <CALL PE,nnnn>
CPO nnnn <CALL PO,nnnn>
CZ nnnn <CALL Z,nnnn>

Appel conditionnel de sous-programme à l'adresse nnnn. Les conditions sont les suivantes :

C → C = 1
M → S = 1
NC → C = 0
NZ → Z = 0
P → S = 0
PE → P = 1
PO → P = 0
Z → Z = 1

CMA <CPL>

Effectue un complément à un sur le contenu de l'accumulateur.

Indicateurs mis à un : H,N.

CMC <CCF>

Complémente l'indicateur C.

Indicateur modifié : C.
Indicateur mis à zéro : N.

CMP M <CP (HL)>

Compare l'accumulateur et le contenu de l'adresse référencée par le registre HL. L'indicateur Z est mis à un si les valeurs sont égales. L'indicateur C est mis à un si l'accumulateur est plus petit que l'opérande.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

CMP r <CP r>

Compare le registre r et l'accumulateur. L'indicateur Z est mis à un

si les valeurs sont égales. L'indicateur C est mis à un si l'accumulateur est plus petit que le registre r.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

CPI nn <CP nn>

Compare la constante nn et l'accumulateur. L'indicateur Z est mis à un si les valeurs sont égales. L'indicateur C est mis à un si l'accumulateur est plus petit que la constante nn.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

DAA <DAA>

Ajustement décimal de l'accumulateur. Est utilisé pour l'addition BCD. Contrairement au Z80, cette instruction donne un résultat erroné pour la soustraction.

Indicateurs modifiés : C, H, O, S, Z.

DAD B <ADD HL,BC>
DAD D <ADD HL,DE>
DAD H <ADD HL,HL>
DAD SP <ADD HL,SP>

Additionne le registre double spécifié comme opérande et le registre double HL. Le registre C est mis à un si le résultat fait plus de 16 bits.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à zéro : N.

DCR M <DEC (HL)>

Décrémente le contenu de l'adresse référencée par le registre HL.

Indicateurs modifiés : H, O, S, Z.
Indicateur mis à un : N.
Indicateur non modifié : C.

DCR r <DEC r>

Décrémente le contenu du registre r.

Indicateurs modifiés : H, O, S, Z.
Indicateur mis à un : N.
Indicateur non modifié : C.

DCX B <DEC BC>
DCX D <DEC DE>
DCX H <DEC HL>
DCX SP <DEC SP>

Décrémente le registre double opérande.

Indicateur modifié : aucun.

DI <DI>

Rend les demandes d'interruption inopérantes.

EI <EI>

Rétablit la prise en compte des demandes d'interruption.

HLT <HALT>

Suspend l'activité du CPU jusqu'à un reset ou une interruption.

IN nn <IN A,(nn)>

Place dans l'accumulateur la valeur présente sur le port d'adresse nn.

Indicateur modifié : aucun.

INR M <INC (HL)>

Incrémente le contenu de l'adresse référencée par le registre HL.

Indicateurs modifiés : H, O, S, Z.
Indicateur mis à un : N.
Indicateur non modifié : C.

INR r <INC r>

Incrémente le registre r.

Indicateurs modifiés : H, O, S, Z.
Indicateur mis à un : N.
Indicateur non modifié : C.

INX B <INC BC>
INX D <INC BE>
INX H <INC HL>
INX SP <INC SP>

Incrémente le registre double spécifié comme opérande.

Indicateur modifié : aucun.

JMP nnnn <JP nnnn>

Saut inconditionnel à l'adresse nnnn.

Indicateur modifié : aucun.

JC nnnn <JP C,nnnn>
JM nnnn <JP M,nnnn>

JNC nnnn <JP NC,nnnn>
JNZ nnnn <JP NZ,nnnn>
JP nnnn <JP P,nnnn>
JPE nnnn <JP PE,nnnn>
JPO nnnn <JP PO,nnnn>
JZ nnnn <JP Z,nnnn>

Saut conditionnel à l'adresse nnnn. Les conditions sont les suivantes :

C → C=1
M → S=1
NC → C=0
NZ → Z=0
P → S=0
PE → P=1
PO → P=0
Z → Z=1

LDA nnnn <LD A, (nnnn)>

Place dans l'accumulateur la valeur se trouvant à l'adresse nnnn.

LDAX B <LD A,(BC)>
LDAX D <LD A,(DE)>

Place dans l'accumulateur la valeur se trouvant à l'adresse référencée par les registres BC ou DE.

LHLD nnnn <LD HL, (nnnn)>

Charge le registre L avec le contenu de l'adresse nnnn et le registre H avec le contenu de l'adresse nnnn + 1.

LXI B,nnnn <LD BC,nnnn>
LXI D,nnnn <LD DE,nnnn>

LXI H,nnnn <LD HL,nnnn>
LXI SP,nnnn <LD SP,nnnn>
Charge le registre double spécifié avec la constante nnnn.

MOV M,r <LD (HL),r>

Place le contenu du registre r à l'adresse référencée par le registre double HL.

MOV r,M <LD r, (HL)>

Place le contenu de l'adresse référencée par le registre double HL dans le registre r.

MOV r,r' <LD r,r'>

Place le contenu du registre r' dans le registre r.

MVI M,nn <LD (HL),nn>

Place la constante nn à l'adresse référencée par le registre HL.

MVI r,nn <LD r,nn>

Place la constante nn dans le registre r.

NOP <NOP>

Aucune opération n'est exécutée.

ORA M <OR (HL)>

Effectue un OU logique entre l'accumulateur et le contenu de l'adresse référencée par le registre HL. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, H, N.

ORA r <OR r>

Effectue un OU logique entre l'accumulateur et le registre r. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, H, N.

ORI nn <OR nn>

Effectue un OU logique entre l'accumulateur et la constante nn. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, H, N.

OUT nn <OUT (nn),A>

Place le contenu de l'accumulateur dans le port d'adresse nn.

PCHL <JP (HL)>

Place le contenu du registre HL dans le compteur de programme (PC). Cette instruction équivaut à un branchement à l'adresse contenue dans HL.

POP B <POP BC>
POP D <POP DE>
POP H <POP HL>

Place les deux premières valeurs de la pile dans le registre double spécifié. La première valeur est placée dans le registre de poids faible (C, E ou L). La seconde valeur est placée dans le registre de poids fort (B, D ou H). Le pointeur de pile (SP) est incrémenté deux fois.

POP PSW <POP AF>

Place la première valeur de la pile dans le registre des indicateurs, et la seconde valeur dans l'accumulateur. Le pointeur de pile (SP) est incrémenté deux fois.

PUSH B <PUSH BC>
PUSH D <PUSH DE>
PUSH H <PUSH HL>

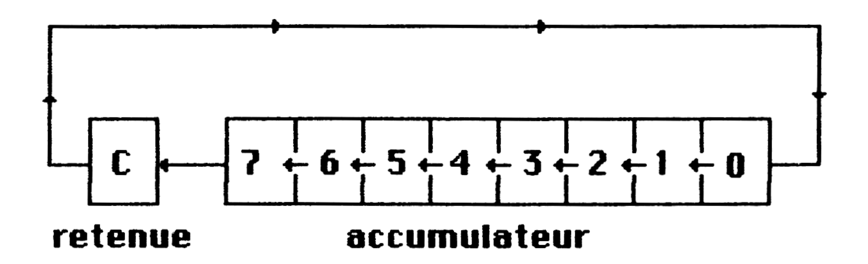
Place le contenu du registre spécifié dans la pile. Le contenu du registre de poids fort (B, D ou H) est d'abord placé sur la pile, puis le contenu du registre de poids faible (C, E, L). Le pointeur de pile (SP) est décrémenté deux fois.

PUSH PSW <PUSH AF>

Place le contenu de l'accumulateur puis le contenu du registre des indicateurs sur la pile. Le pointeur de pile (SP) est décrémenté deux fois.

RAL <RLA>

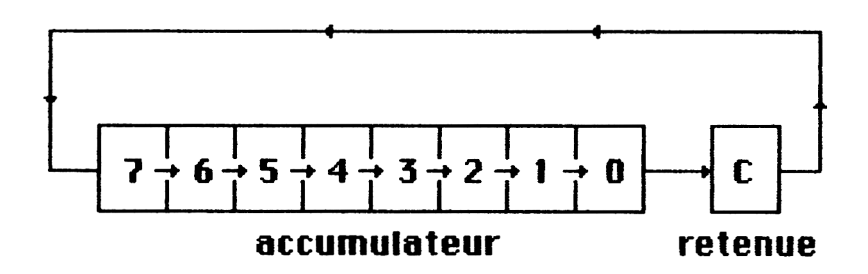
Effectue une rotation vers la gauche du contenu de l'accumulateur selon le schéma ci-dessous.



Indicateur modifié : C.
Indicateurs mis à zéro : H, N.

RAR <RRA>

Effectue une rotation vers la droite du contenu de l'accumulateur selon le schéma ci-dessous.



Indicateur modifié : C.
Indicateurs mis à zéro : H, N.

RET <RET>

Retour de sous-programme. Les deux premières valeurs de la pile sont placées dans le compteur de programme (PC). Le pointeur de pile est incrémenté deux fois.

RC	<RET C>
RM	<RET M>
RNC	<RET NC>
RNZ	<RET NZ>
RP	<RET P>
RPE	<RET PE>
RPO	<RET PO>
RZ	<RET Z>

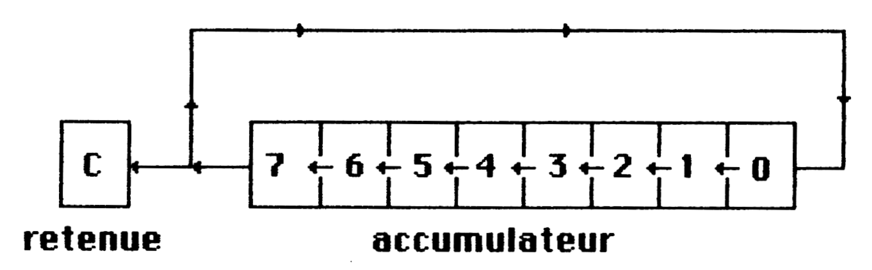
Retour conditionnel de sous-programme. Si la condition est remplie, les deux premières valeurs de la pile sont placées dans le compteur de programme (PC). Le pointeur de pile est incrémenté deux fois. Les conditions sont :

C → C=1
M → S=1

NC → C=0
NZ → Z=0
P → S=0
PE → P=1
PO → P=0
Z → Z=1

RLC <RLCA>

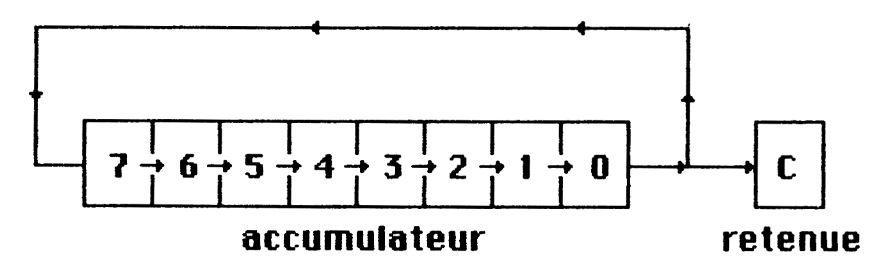
Effectue une rotation vers la gauche du contenu de l'accumulateur selon le schéma ci-dessous.



Indicateur modifié : C
Indicateurs mis à zéro : H, N.

RRC <RRCA>

Effectue une rotation vers la droite du contenu de l'accumulateur selon le schéma ci-dessous.



Indicateur modifié : C.
Indicateurs mis à zéro : H, N.

RST 0 <RST 00H>
RST 1 <RST 08H>
RST 2 <RST 10H>
RST 3 <RST 18H>
RST 4 <RST 20H>
RST 5 <RST 28H>
RST 6 <RST 30H>
RST 7 <RST 38H>

Ces instructions génèrent un appel de sous-programme aux adresses 00H, 08H, 10H, 18H, 20H, 28H, 30H ou 38H.

SBB M <SBC A,(HL)>

Soustrait le contenu de l'indicateur C et le contenu de l'adresse mémoire référencée par le registre HL de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

SBB r <SBC A,r>

Soustrait le contenu de l'indicateur C et le registre r de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

SBI nn <SBC A,nn>

Soustrait la constante nn et le contenu de l'indicateur C de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

SHLD nnnn <LD (nnnn),HL>

Place le contenu du registre L à l'adresse nnnn et le contenu du registre H à l'adresse nnnn+1.

SPHL <LD SP,HL>

Place le contenu du registre HL dans le registre SP.

STA nnnn <LD (nnnn),A>

Place le contenu de l'accumulateur à l'adresse nnnn.

STAX B <LD (BC),A>
STAX D <LD (DE),A>

Place le contenu de l'accumulateur dans l'adresse référencée par le registre double spécifié.

STC <SCF>

Met l'indicateur C à 1.

Indicateur mis à un : C.
Indicateurs mis à zéro : H, N.

SUB M <SUB (HL)>

Soustrait le contenu de l'adresse référencée par HL de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

SUB r <SUB r>

Soustrait le registre r de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

SUI nn <SUB nn>

Soustrait la constante nn de l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : C, H, O, S, Z.
Indicateur mis à un : N.

XCHG <EX DE,HL>

Échange les contenus des registres DE et HL.

Indicateur modifié : aucun.

XRA M <XOR (HL)>

Effectue un OU exclusif entre l'accumulateur et la valeur se trouvant à l'adresse référencée par le registre double HL. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, H, N.

XRA r <XOR r>

Effectue un OU exclusif entre l'accumulateur et le registre r. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, H, N.

XRI nn <XOR nn>

Effectue un OU exclusif entre la constante nn et l'accumulateur. Le résultat est placé dans l'accumulateur.

Indicateurs modifiés : P, S, Z.
Indicateurs mis à zéro : C, H, N.

XTHL <EX (SP),HL>

Échange le contenu de l'adresse référencée par le pointeur de pile SP et le registre L. Échange le contenu de l'adresse référencée par SP+1 et le registre H.

Indicateur modifié : aucun.

6
LES COMMANDES DE ED

LES CARACTÈRES DE CONTROLE

↑C	Démarrage à chaud. Sortie de ED.
↑E	Fin de ligne physique. Permet de continuer une commande sur la ligne suivante.
↑H	Retour arrière avec effacement d'un caractère.
↑I	Tabulation. Identique à la touche TAB.
↑J	Fin de ligne logique. Identique à la touche ENTER.
↑L	Représente une fin de ligne dans les chaînes de caractères utilisées avec les commandes de recherche et de remplacement.
↑M	Fin de ligne logique. Identique à la touche ENTER.
↑P	Déclenche et arrête l'écho de l'affichage écran sur l'imprimante.
↑R	Réaffiche la ligne en cours avec les modifications effectuées.
↑S	Suspend l'affichage temporairement. L'affichage reprend après la frappe d'une touche quelconque.
↑U	Efface la ligne en cours.
↑X	Efface la ligne en cours.
↑Z	Fin d'insertion. Séparateur de chaînes de caractères dans les commandes de recherche et de remplacement.

LES COMMANDES

nA	Charge n lignes en mémoire.

+/—B	Place le pointeur au début ou à la fin du texte en mémoire.
+/—nC	Déplace le pointeur de n caractères.
+/—nD	Détruit n caractères.
E	Quitte l'éditeur en sauvegardant les modifications.
nF	Recherche une chaîne de caractères.
H	Sauvegarde les modifications et ramène le pointeur au début du fichier.
I	Insère du texte.
nJ	Juxtapose des chaînes de caractères.
+/—nK	Détruit n lignes de texte.
+/—nL	Déplace le pointeur de n lignes.
nM	Exécute n fois une macro-commande.
nN	Recherche la $n^{ième}$ occurrence d'une chaîne de caractères.
O	Revient au fichier original.
Q	Quitte l'éditeur sans sauvegarder les modifications.
R	Lit un fichier sur une disquette et l'insère dans le texte en mémoire.
nS	Substitution de chaînes de caractères.
+/—nT	Affiche n lignes de texte.
+/—U	Transforme les majuscules en minuscules.
+/—/0V	Supprime ou rétablit les numéros de lignes. 0V affiche la mémoire totale et la mémoire libre.
nW	Écrit n lignes sur la disquette.
nZ	Arrête l'éditeur pendant un temps variable.
+/—n	Déplace le pointeur de n lignes et affiche la ligne contenant le pointeur.
n:	Place le pointeur sur la ligne n.

7
LES PARAMÈTRES DE PIP

LES NOMS D'UNITÉS LOGIQUES

CON:	Console. Unité d'entrée et de sortie.
RDR:	Unité d'entrée.
PUN:	Unité de sortie.
LST:	Unité de sortie (listage).

LES NOMS D'UNITÉS PHYSIQUES

TTY:	Premier dispositif d'entrée/sortie (Interface série). Généralement affectée à RDR: en entrée et à PUN: en sortie.
BAT:	Équivalente en entrée de RDR: et en sortie de LST:.
CRT:	Écran/clavier. Généralement affectée à CON:.
PTR:	Entrée. Envoie un caractère "fin de fichier".
PTP:	Sortie factice. Utilisée pour tester des programmes.
LPT:	Interface Centronics. Généralement affectée à LST:.
UC1:	Second dispositif d'entrée/sortie.
UR1:	Second dispositif d'entrée/sortie (en entrée seulement).
UR2:	Clavier.
UP1:	Second dispositif d'entrée/sortie (en sortie seulement).
UP2:	Écran.
UL1:	Second dispositif d'entrée/sortie.

LES UNITÉS PARTICULIÈRES DE PIP

NUL:	Envoie quarante caractères NUL (ASCII 0).
EOF:	Envoie un caractère ↑Z.
PRN:	Identique à LST: mais les lignes sont numérotées, les tabulations sont étendues, et un saut de page est inséré toutes les soixante lignes.
INP:	Unité d'entrée correspondant à une zone de PIP qui peut être modifiée par l'utilisateur.
OUT:	Unité de sortie correspondant à une zone de PIP qui peut être modifiée par l'utilisateur.

LES PARAMÈTRES

B	Mode de transfert par blocs.
Dn	Effacement des caractères après la colonne n.
E	Affichage des données sur l'écran pendant le transfert.
F	Suppression des sauts de page.
Gn	Transfert depuis la zone utilisateur n.
H	Vérification des données selon le format hexadécimal de Intel.
I	Suppression des enregistrements ":00" dans les fichiers au format hexadécimal.
L	Transformation de toutes les majuscules en minuscules.
N	Numérotation des lignes.
O	Copie de fichier binaire ; les caractères "fin de texte" (↑Z) sont ignorés dans les concaténations.
Pn	Ajout d'un saut de page toutes les n lignes.
Qchaîne↑Z	Transfert jusqu'à la première occurrence de la chaîne de caractères spécifiée.
R	Transfert des fichiers système.

Schaîne↑Z	Transfert à partir de la première occurrence de la chaîne spécifiée.
Tn	Extension des tabulations à n caractères.
U	Transformation des caractères minuscules en majuscules.
V	Vérification des données transférées.
W	Transfert avec écriture des fichiers protégés.
Z	Force le bit de parité des données transférées à zéro.

8
LES COMMANDES DE CP/M

La liste ci-dessous donne toutes les commandes de CP/M avec l'indication de leur type. La troisième colonne indique s'il s'agit de commandes CP/M standard ou de commandes particulières à l'Amstrad. (Il faut noter que les commandes particulières à l'Amstrad peuvent exister sur d'autres systèmes, sous le même nom.)

AMSDOS	non résidente	Amstrad
ASM	non résidente	CP/M
BOOTGEN	non résidente	Amstrad
CHKDISC	non résidente	Amstrad
CLOAD	non résidente	Amstrad
COPYDISC	non résidente	Amstrad
CSAVE	non résidente	Amstrad
DDT	non résidente	CP/M
DIR	résidente	CP/M
DISCCOPY	non résidente	Amstrad
DISCCHK	non résidente	Amstrad
DISCKIT2	non résidente	Amstrad
DRLKEYS	non résidente	Amstrad
DUMP	non résidente	CP/M
ED	non résidente	CP/M
ERA	résidente	CP/M
FILECOPY	non résidente	Amstrad
FORMAT	non résidente	Amstrad
FWRESET	non résidente	Amstrad
LOAD	non résidente	CP/M
MOVCPM	non résidente	CP/M
PIP	non résidente	CP/M
REN	résidente	CP/M
SAVE	résidente	CP/M
SETUP	non résidente	Amstrad
STAT	non résidente	CP/M

SUBMIT	non résidente	CP/M
SYSGEN	non résidente	CP/M
TYPE	résidente	CP/M
USER	résidente	CP/M
XSUB	non résidente	CP/M

9

LES EXTENSIONS

ASM	Fichier source en assembleur.
BAK	Fichier de sécurité produit par ED.
BAS	Programme BASIC.
BIN	Fichier contenant un programme ou l'image d'une zone mémoire sauvegardés sous AMSDOS.
COM	Commande non résidente.
HEX	Fichier hexadécimal produit par la commande LOAD.
PRN	Listing d'assemblage produit par l'assembleur.
SUB	Fichier de commandes destiné à la commande SUBMIT.
$$$	Fichier temporaire.

10
LES CODES ÉCRAN

Valeur	Paramètres	Effet
00H		Aucun
01H	0 à 255	Affichage du caractère dont le code ASCII est le paramètre. Il est ainsi possible d'afficher les caractères 0 à31.
02H		Suppression du curseur texte. Équivalent de la commande BASIC CURSOR 0.
03H		Affichage du curseur texte. Équivalent de la commande BASIC CURSOR 1.
04H	0 à 2	Sélection du mode écran. Équivalent de la commande BASIC MODE.
05H	0 à 255	Passage du paramètre au curseur graphique.
06H		Activation de l'écran texte.
07H		Émission d'un BIP sonore.
08H		Déplacement du curseur d'un caractère vers la gauche.
09H		Déplacement du curseur d'un caractère vers la droite.
0AH		Déplacement du curseur d'une ligne vers le bas.
0BH		Déplacement du curseur d'une ligne vers le haut.
0CH		Effacement de la fenêtre. Le curseur est renvoyé dans le coin supérieur gauche. Équivalent de la commande BASIC CLS.

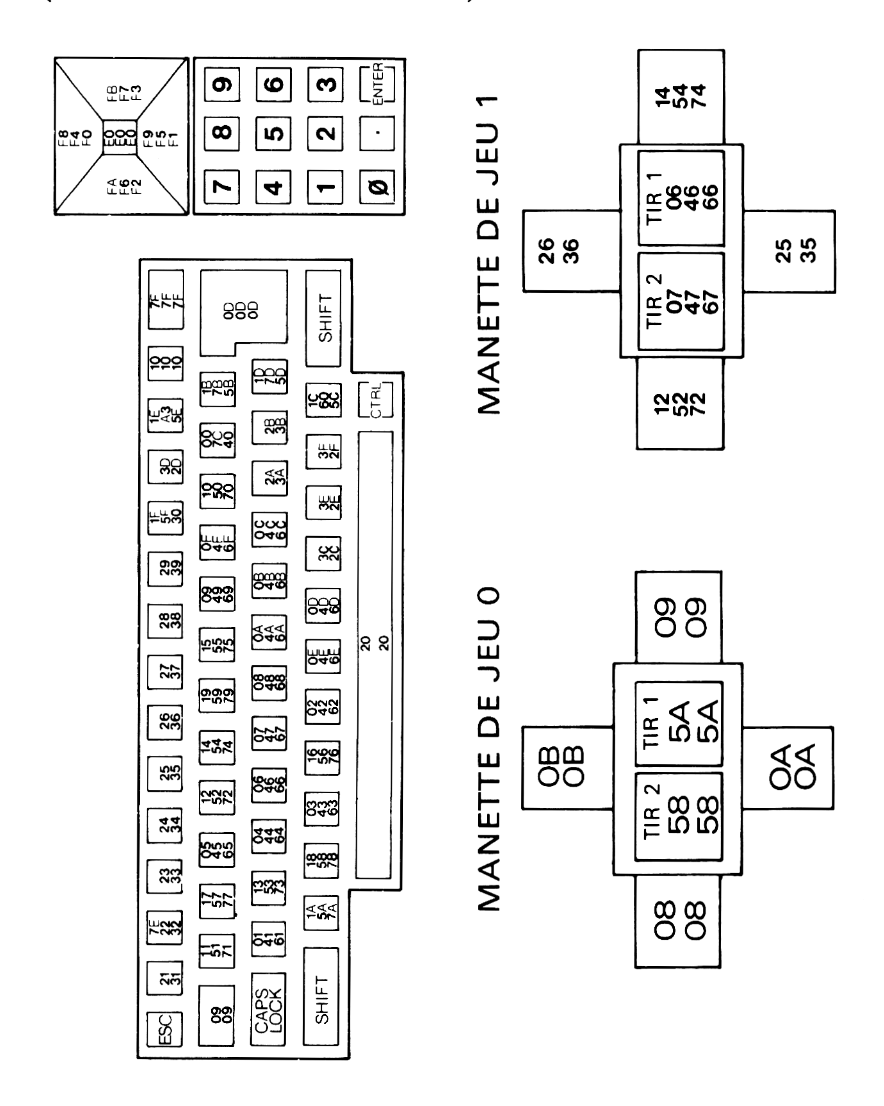
Valeur	Paramètres	Effet
0DH		Renvoi du curseur en début de ligne.
0EH	0 à 15	Choix de la couleur du fond. La couleur est prise modulo 16. Équivalent de la commande BASIC PAPER.
0FH	0 à 15	Choix de la couleur des caractères. La couleur est prise modulo 16. Équivalent de la commande BASIC PEN.
10H		Effacement d'un caractère à la position du curseur.
11H		Effacement du début de la ligne jusqu'à la position du curseur.
12H		Effacement de la fin de la ligne à partir de la position du curseur.
13H		Effacement de la fenêtre jusqu'à la position du curseur.
14H		Effacement de la fenêtre à partir de la position du curseur.
15H		Inhibe l'écran texte. Plus rien ne sera affiché jusqu'à réception d'un caractère 06H.
16H	0 ou 1	Active (1) ou inhibe (0) le mode transparent. Le paramètre est pris modulo 2.
17H	0 à 3	0=Mode graphique normal. 1=Mode XOR. 2=Mode AND. 3=Mode OR. Le paramètre est pris modulo 4.
18H		Échange des couleurs de fond et de caractères.

Valeur	Paramètres	Effet
19H	0 à 255 0 à 255 0 à 255 0 à 255 0 à 255 0 à 255 0 à 255 0 à 255 0 à 255	Définition d'un caractère. Équivalent de la commande BASIC SYMBOL. Le premier des neuf paramètres est le code du caractère à définir. Les huit paramètres suivants définissent la matrice du caractère.
1AH	1 à 80 1 à 80 1 à 25 1 à 25	Définition d'une fenêtre. Équivalent de la commande BASIC WINDOW. Les deux premiers paramètres définissent les côtés gauche et droit de la fenêtre. Les deux suivants déterminent le haut et le bas de la fenêtre.
1BH		Aucun.
1CH	0 à 15 0 à 31 0 à 31	Choix d'une couleur d'encre. Équivalent de la commande BASIC INK. Le premier paramètre (modulo 16) est le numéro de l'encre. Les deux autres (modulo 32) sont les deux couleurs utilisées alternativement. (Les couleurs 27 à 31 ne sont pas définies.)
1DH	0 à 31 0 à 31	Fixe les couleurs de la bordure. Équivalent de la commande BASIC BORDER. Les deux paramètres (modulo 32) sont les deux couleurs à utiliser alternativement.
1EH		Place le curseur dans le coin supérieur gauche de la fenêtre.
1FH	1 à 80	Place le curseur à une position donnée. Équivalent de la commande BASIC LOCATE. Le premier paramètre est la colonne et le second est la ligne.

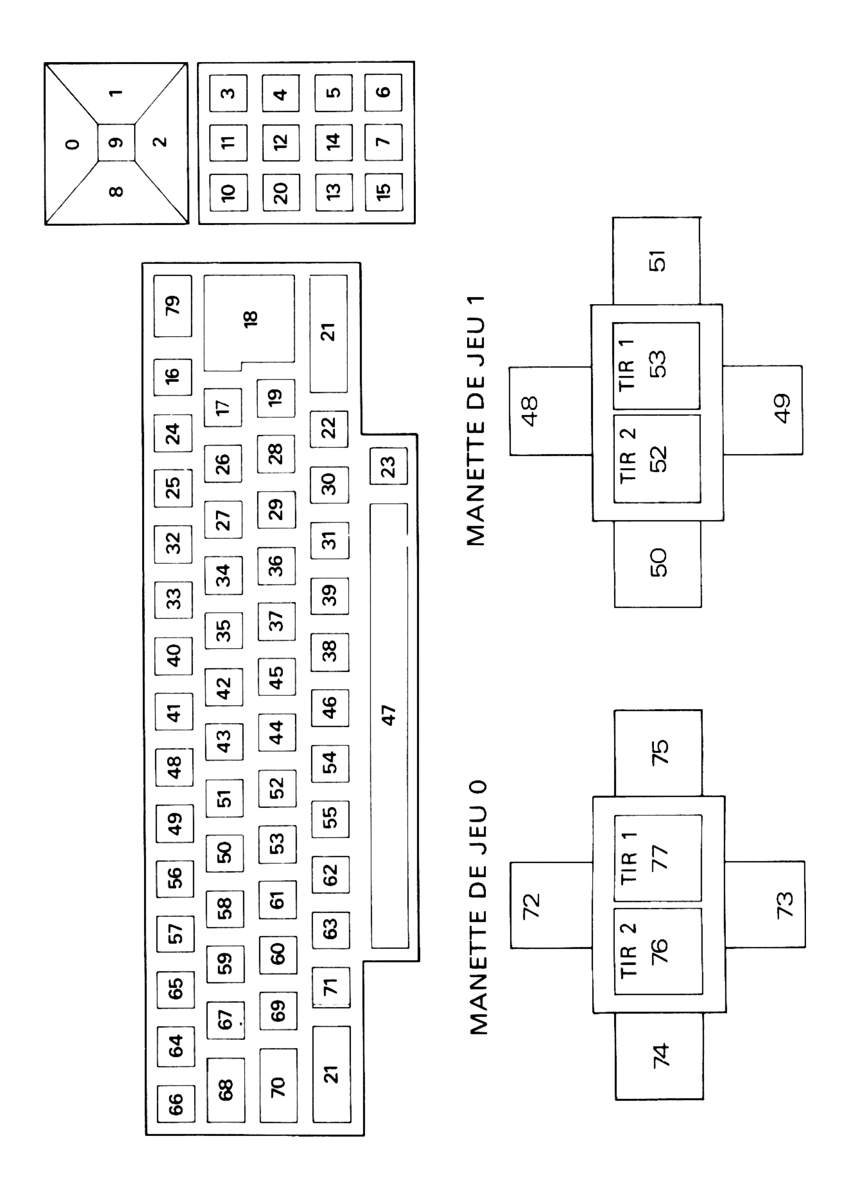
11
LES CODES CLAVIER

LES CODES ASCII

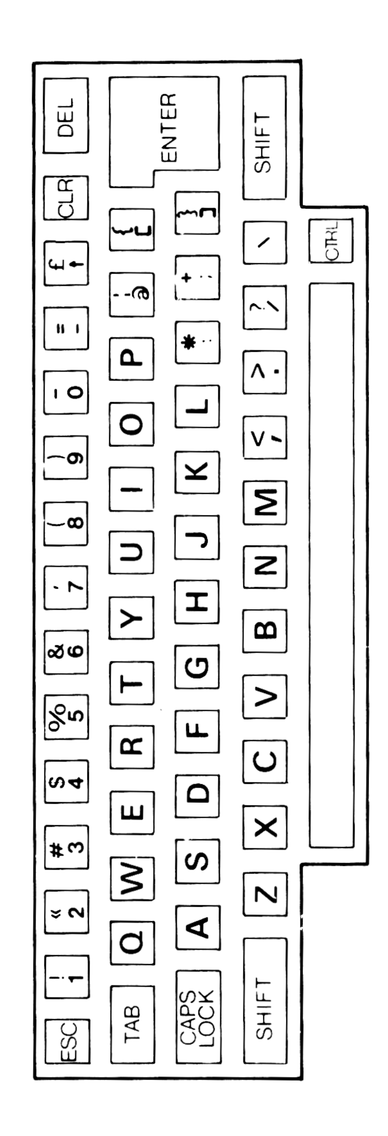
(Les codes sont en hexadécimal.)

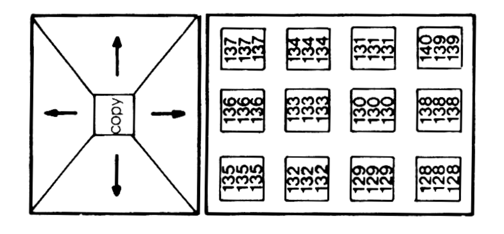


(Les numéros sont en décimal.)



(Les valeurs sont en décimal.)





12
LES CODES COULEURS

0	NOIR
1	BLEU
2	BLEU VIF
3	ROUGE
4	MAGENTA
5	MAUVE
6	ROUGE VIF
7	POURPRE
8	MAGENTA VIF
9	VERT
10	TURQUOISE
11	BLEU CIEL
12	JAUNE
13	BLANC
14	BLEU PASTEL
15	ORANGE
16	ROSE
17	MAGENTA PASTEL
18	VERT VIF
19	VERT MARIN
20	TURQUOISE VIF
21	VERT CITRON
22	VERT PASTEL
23	TURQUOISE PASTEL
24	JAUNE VIF
25	JAUNE PASTEL
26	BLANC BRILLANT

LA BIBLIOTHÈQUE SYBEX

OUVRAGES GÉNÉRAUX

VOTRE PREMIER ORDINATEUR *par Rodnay Zaks,*
296 pages, Réf. 394

VOTRE ORDINATEUR ET VOUS *par Rodnay Zaks,*
296 pages, Réf. 271

DU COMPOSANT AU SYSTÈME : une introduction aux microprocesseurs *par Rodnay Zaks,*
636 pages, Réf. 0040

TECHNIQUES D'INTERFACE aux microprocesseurs *par Austin Lesea et Rodnay Zaks,*
450 pages, Réf. 0039

LEXIQUE INTERNATIONAL MICRO-ORDINATEURS, avec dictionnaire abrégé en 10 langues
192 pages, Réf. 234

GUIDE DES MICRO-ORDINATEURS A MOINS 3 000 F *par Joël Poncet,*
144 pages, Réf. 322

LEXIQUE MICRO-INFORMATIQUE *par Pierre Le Beux,*
140 pages, Réf. 369

LA SOLUTION RS-232 *par Joe Campbell,*
208 pages, Réf. 0052

MINITEL ET MICRO-ORDINATEUR *par Pierrick Bourgault,*
198 pages, Réf. 0119

ROBOTS - CONSTRUCTION, PROGRAMMATION *par Fernand Esteves,*
400 pages, Réf. 0130

ALGORITHMES *par Pierre Beaufils et Wolfram Luther,*
296 pages, Réf. 0149

BASIC

VOTRE PREMIER PROGRAMME BASIC *par Rodnay Zaks,*
208 pages, Réf. 263

INTRODUCTION AU BASIC *par Pierre Le Beux,*
336 pages, Réf. 0035

LE BASIC PAR LA PRATIQUE : 60 exercices *par Jean-Pierre Lamoitier,*
252 pages, Réf. 0095

LE BASIC POUR L'ENTREPRISE *par Xuan Tung Bui,*
204 pages, Réf. 253

PROGRAMMES EN BASIC, Mathématiques, Statistiques, Informatique *par Alan R. Miller,*
318 pages, Réf. 259

BASIC, PROGRAMMATION STRUCTURÉE *par Richard Mateosian,*
352 pages, Réf. 0129

JEUX D'ORDINATEUR EN BASIC *par David H. Ahl,*
192 pages, Réf. 246

NOUVEAUX JEUX D'ORDINATEUR EN BASIC *par David H. Ahl,*
204 pages, Réf. 247

FICHIERS EN BASIC *par Alan Simpson,*
256 pages, Réf. 0102

TECHNIQUES DE PROGRAMMATION EN BASIC *par S. Crosmarie, M. Perron et D. Philippine*
152 pages, Réf. 0124

PASCAL

INTRODUCTION AU PASCAL *par Pierre Le Beux,*
496 pages, Réf. 0030

LE PASCAL PAR LA PRATIQUE *par Pierre Le Beux et Henri Tavernier,*
562 pages, Réf. 361

LE GUIDE DU PASCAL *par Jacques Tiberghien,*
504 pages, Réf. 423

PROGRAMMES EN PASCAL pour Scientifiques et Ingénieurs *par Alan R. Miller,*
392 pages, Réf. 240

AUTRES LANGAGES

INTRODUCTION A ADA *par Pierre Le Beux,*
366 pages, Réf. 360

INTRODUCTION A C *par Bruce Hunter,*
312 pages, Réf. 0092

MICRO-ORDINATEURS

ALICE

JEUX EN BASIC POUR ALICE *par Pierre Monsaut,*
96 pages, Réf. 320

ALICE et ALICE 90, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 376

ALICE, 56 PROGRAMMES *par Stanley R. Trost,*
160 pages, Réf. 401

ALICE, GUIDE DE L'UTILISATEUR *par Norbert Rimoux,*
208 pages, Réf. 378

ALICE, PROGRAMMATION EN ASSEMBLEUR *par Georges Fagot-Barraly,*
192 pages, Réf. 420

AMSTRAD

AMSTRAD, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 0105

AMSTRAD, 56 PROGRAMMES *par Stanley R. Trost,*
160 pages, Réf. 0107

AMSTRAD, JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 0108

AMSTRAD, PROGRAMMATION EN ASSEMBLEUR
par Georges Fagot-Barraly,
208 pages, Réf. 0136

AMSTRAD EXPLORÉ *par John Braga,*
192 pages, Réf. 0135

AMSTRAD, GUIDE DU GRAPHISME *par James Wynford,*
208 pages, Réf. 0141

AMSTRAD CP/M 2.2 *par Anatole D'Hardencourt,*
248 pages, Réf. 0156

AMSTRAD ASTROLOGIE/NUMEROLOGIE/BIORYTHMES
par Pierrick Bourgault,
160 pages, Réf. 0167

AMSTRAD MULTIPLAN *de Microsoft,*
496 pages, Réf. 1111

AMSTRAD, CRÉER DE NOUVELLES INSTRUCTIONS
par Jean-Claude Despoine,
144 pages, Réf. 0176

AMSTRAD ASTROCALC
par Gérard Blanc et Philippe Destrebecq,
168 pages, Réf. 0162

APPLE / MACINTOSH

PROGRAMMEZ EN BASIC SUR APPLE II,
Tomes 1 et 2 *par Léopold Laurent,*
208 pages, Réf. 333 et 380

APPLE II 66 PROGRAMMES BASIC *par Stanley R. Trost,*
192 pages, Réf. 283

JEUX EN PASCAL SUR APPLE
par Douglas Hergert et Joseph T. Kalash,
372 pages, Réf. 241

GUIDE DU BASIC APPLE II *par Douglas Hergert,*
272 pages, Réf. 0006

APPLE II, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 373

MACINTOSH, GUIDE DE L'UTILISATEUR
par Joseph Caggiano,
208 pages, Réf. 396

APPLE IIC, GUIDE DE L'UTILISATEUR
par Thomas Blackadar,
160 pages, Réf. 0089

MULTIPLAN SUR MACINTOSH
par Goulven Habasque,
240 pages, Réf. 0099

INTRODUCTION A MAC PASCAL *par Pierre Le Beux,*
416 pages, Réf. 0145

MACINTOSH POUR LA PRESSE, L'EDITION ET
LA PUBLICITE *par Bernard Le Du,*
160 pages, Réf. 0173

ATARI

JEUX EN BASIC SUR ATARI *par Paul Bunn,*
96 pages, Réf. 282

ATARI, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 387

ATARI, GUIDE DE L'UTILISATEUR *par Thomas Blackadar,*
192 pages, Réf. 354

ATMOS

JEUX EN BASIC SUR ATMOS *par Pierre Monsaut,*
96 pages, Réf. 346

ATMOS, 56 PROGRAMMES *par Stanley R. Trost,*
180 pages, Réf. 372

COMMODORE 64

JEUX EN BASIC SUR COMMODORE 64
par Pierre Monsaut,
96 pages, Réf. 0017

COMMODORE 64, PREMIERS PROGRAMMES
par Rodnay Zaks,
248 pages, Réf. 342

GUIDE DU BASIC VIC 20, COMMODORE 64
par Douglas Hergert,
240 pages, Réf. 312

COMMODORE 64, GUIDE DE L'UTILISATEUR
par J. Kascmer,
144 pages, Réf. 314

COMMODORE 64, 66 PROGRAMMES
par Stanley R. Trost,
192 pages, Réf. 319

COMMODORE 64, GUIDE DU GRAPHISME
par Charles Platt,
372 pages, Réf. 0053

COMMODORE 64, JEUX D'ACTION *par Eric Ravis,*
96 pages, Réf. 403

COMMODORE 64, 1ers CONTACTS
par Marty Dejonghe et Caroline Earhart,
208 pages, Réf. 390

COMMODORE 64, BASIC APPROFONDI
par Gary Lippman,
216 pages, Réf. 0100

DRAGON

JEUX EN BASIC SUR DRAGON *par Pierre Monsaut,*
96 pages, Réf. 324

EXL 100

EXL 100, JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 0126

GOUPIL

PROGRAMMEZ VOS JEUX SUR GOUPIL
par François Abella,
208 pages, Réf. 264

HECTOR

HECTOR JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 388

IBM

IBM PC EXERCICES EN BASIC *par Jean-Pierre Lamoitier,*
256 pages, Réf. 338

IBM PC GUIDE DE L'UTILISATEUR
par Joan Lasselle et Carol Ramsey,
160 pages, Réf. 301

IBM PC 66 PROGRAMMES BASIC *par Stanley R. Trost,*
192 pages, Réf. 359

GRAPHIQUES SUR IBM PC *par Nelson Ford,*
320 pages, Réf. 357

GUIDE DE PC DOS *par Richard A. King,*
240 pages, Réf. 0013

LASER

LASER JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 371

MO 5

MO 5 JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 0067

MO 5, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 370

MO 5, 56 PROGRAMMES *par Stanley R. Trost,*
160 pages, Réf. 375

MO 5, PROGRAMMATION EN ASSEMBLEUR
par Georges Fagot-Barraly,
192 pages, Réf. 384

MO 5, DYNAMIQUE CINÉMATIQUE, MÉTHODE POUR LA PROGRAMMATION DES JEUX *par Daniel Lebigre,*
272 pages, Réf. 0118

MO 5, STATIQUE, DYNAMIQUE, ELECTRONIQUE, PROGRAMMES DE PHYSIQUE EN BASIC
par Beaufils, Lamarche et Muggianu,
240 pages, Réf. 0148

MO 5, PROGRAMMES D'ELECTRONIQUE EN BASIC
par Beaufils, Delusurieux, Do, Romanacce,
312 pages, Réf. 0143

MO 5, OPTIQUE, THERMODYNAMIQUE, CHIMIE
par P. Beaufils, M. Lamarche, Y. Muggianu,
224 pages, Réf. 0161

MSX

MSX, JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 411

MSX, INITIATION AU BASIC *par Rodnay Zaks,*
248 pages, Réf. 410

MSX, 56 PROGRAMMES *par Stanley R. Trost,*
160 pages, Réf. 0109

MSX, GUIDE DU GRAPHISME *par Mike Shaw,*
192 pages, Réf. 0132

MSX, PROGRAMMES EN LANGAGE MACHINE
par Steeve Webb,
112 pages, Réf. 0153

MSX, PROGRAMMATION EN ASSEMBLEUR
par Georges Fagot-Barraly,
216 pages, Réf. 0144

MSX, GUIDE DU BASIC *par Michel Laurent,*
264 pages, Réf. 0155

MSX, JEUX EN ASSEMBLEUR *par Eric Ravis*
112 pages, Réf. 0170

MSX, ROUTINES GRAPHIQUES EN ASSEMBLEUR
par Steeve Webb
88 pages, Réf. 0154

MSX, TECHNIQUES DE PROGRAMMATION DES JEUX EN ASSEMBLEUR
par Georges Fagot-Barraly,
176 pages, Réf. 0178

MSX ASTROLOGIE/NUMEROLOGIE/BIORYTHMES
par Pierrick Bourgault,
157 pages, Réf. 0168

ORIC

JEUX EN BASIC SUR ORIC *par Peter Shaw,*
96 pages, Réf. 278

ORIC PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 344

SHARP

DÉCOUVREZ LE SHARP PC-1500 ET LE TRS-80 PC-2
par Michel Lhoir,
2 tomes, Réf. 261-262

SPECTRAVIDEO

SPECTRAVIDEO, JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 377

SPECTRUM

PROGRAMMEZ EN BASIC SUR SPECTRUM
par S.M. Gee,
208 pages, Réf. 252

JEUX EN BASIC SUR SPECTRUM *par Peter Shaw,*
96 pages, Réf. 276

SPECTRUM, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 381

SPECTRUM JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 368

TI 99/4

PROGRAMMEZ VOS JEUX SUR TI 99/4
par François Abella,
160 pages, Réf. 303

TO 7

JEUX EN BASIC SUR TO 7 *par Pierre Monsaut,*
96 pages, Réf. 0026

TO 7, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 328

TO 7, PROGRAMMATION EN ASSEMBLEUR
par Georges Fagot-Barraly,
192 pages, Réf. 350

JEUX SUR TO 7 et MO 5 *par Georges Fagot-Barraly,*
168 pages, Réf. 0134

GESTION DE FICHIERS SUR TO 7 ET MO 5
par Jean-Pierre Lhoir,
136 pages, Réf. 0127

TO 7, 56 PROGRAMMES *par Stanley R. Trost,*
160 pages, Réf. 374

TO 7 / MO 5, GUIDE DU BASIC
par Jean-Louis Greco et Michel Laurent,
288 pages, Réf. 0158

TO 7 / MO 5, GUIDE DU GRAPHISME
par Michel Lamarche et Yves Muggianu,
240 pages, Réf. 0172

TO 7 / MO 5 ASTROLOGIE/NUMEROLOGIE/BIORYTHMES
par Pierrick Bourgault,
160 pages, Réf. 0169

TRS-80

PROGRAMMEZ EN BASIC SUR TRS-80
par Léopold Laurent,
2 tomes, Réf. 366-251

JEUX EN BASIC SUR TRS-80 MC-10 *par Pierre Monsaut,*
96 pages, Réf. 323

JEUX EN BASIC SUR TRS-80 *par Chris Palmer,*
96 pages, Réf. 302

JEUX EN BASIC SUR TRS-80 COULEUR
par Pierre Monsaut,
96 pages, Réf. 325

TRS-80 MODÈLE 100, GUIDE DE L'UTILISATEUR
par Orson Kellog,
112 pages, Réf. 300

TRS-80 COULEUR, PREMIERS PROGRAMMES
par Rodnay Zaks,
248 pages, Réf. 414

TRS-80 COULEUR, 56 PROGRAMMES
par Stanley R. Trost,
160 pages, Réf. 413

VIC 20

PROGRAMMEZ EN BASIC SUR VIC 20
par G. O. Hamann,
2 tomes, Réf. 329-337

JEUX EN BASIC SUR VIC 20 *par Alastair Gourlay,*
96 pages, Réf. 277

VIC 20, PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 341

VIC 20 JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 345

VG 5000

VG 5000, JEUX D'ACTION *par Pierre Monsaut,*
96 pages, Réf. 422

VG 5000, 56 PROGRAMMES *par Stanley R. Trost,*
160 pages, Réf. 0128

ZX 81

ZX 81 GUIDE DE L'UTILISATEUR *par Douglas Hergert,*
208 pages, Réf. 351

ZX 81 56 PROGRAMMES BASIC *par Stanley R. Trost,*
192 pages, Réf. 304

GUIDE DU BASIC ZX 81 *par Douglas Hergert,*
204 pages, Réf. 285

JEUX EN BASIC SUR ZX 81 *par Mark Charlton,*
96 pages, Réf. 275

ZX 81 PREMIERS PROGRAMMES *par Rodnay Zaks,*
248 pages, Réf. 343

MICROPROCESSEURS

PROGRAMMATION DU Z80 *par Rodnay Zaks,*
618 pages, Réf. 0058

APPLICATIONS DU Z80 *par James W. Coffron,*
304 pages, Réf. 0181

PROGRAMMATION DU 6502 *par Rodnay Zaks,*
376 pages, Réf. 0031, 2ème édition

APPLICATIONS DU 6502 *par Rodnay Zaks,*
288 pages, Réf. 332

PROGRAMMATION DU 6800
par Daniel-Jean David et Rodnay Zaks,
374 pages, Réf. 327

PROGRAMMATION DU 6809
par Rodnay Zaks et William Labiak,
392 pages, Réf. 0139

PROGRAMMATION DU 8086/8088
par James W. Coffron,
304 pages, Réf. 0016

MISE EN OEUVRE DU 68000 *par C. Vieillefond,*
352 pages, Réf. 0133

ASSEMBLEUR DU 8086/8088 *par François Retoreau,*
616 pages, Réf. 0093

SYSTÈMES D'EXPLOITATION

GUIDE DU CP/M AVEC MP/M *par Rodnay Zaks,*
354 pages, Réf. 336

CP/M APPROFONDI *par Alan R. Miller,*
380 pages, Réf. 334

INTRODUCTION AU p-SYSTEM UCSD
par Charles W. Grant et Jon Butah,
308 pages, Réf. 365

GUIDE DE MS-DOS *par Richard A. King,*
360 pages, Réf. 0117

INTRODUCTION A UNIX *par John D. Halamka,*
240 pages, Réf. 0098

GUIDE DE PRODOS
par Pierre Beaufils et Wolfram Luther,
248 pages, Réf. 0146

APPLICATIONS ET LOGICIELS

INTRODUCTION AU TRAITEMENT DE TEXTE
par Hal Glatzer,
228 pages, Réf. 243

INTRODUCTION A WORDSTAR *par Arthur Naiman,*
200 pages, Réf. 0062

WORDSTAR APPLICATIONS *par Julie Anne Arca,*
320 pages, Réf. 0005

VISICALC APPLICATIONS *par Stanley R. Trost,*
304 pages, Réf. 258

VISICALC POUR L'ENTREPRISE *par Dominique Helle,*
304 pages, Réf. 309

INTRODUCTION A dBASE II *par Alan Simpson,*
280 pages, Réf. 0064

DE VISICALC A VISI ON *par Jacques Bourdeu,*
256 pages, Réf. 321

MULTIPLAN POUR L'ENTREPRISE
par D. Helle et G. Boussand,
304 pages, Réf. 0079

dBASE II APPLICATIONS *par Christophe Stehly,*
248 pages, Réf. 416

INTRODUCTION A LOTUS 1-2-3
par Chris Gilbert et Laurie Williams,
272 pages, Réf. 0106

INTRODUCTION A dBASE III *par Alan Simpson,*
272 pages, Réf. 0131

LOTUS 1-2-3 POUR L'ENTREPRISE
par Dominique Helle et Guy Boussand,
256 pages, Réf. 0147

LOTUS 1-2-3 PROGRAMMATION DES MACRO-COMMANDES *par Goulven Habasque,*
144 pages, Réf. 0150 F

LOGISTAT, ANALYSE STATISTIQUE DES DONNÉES
par Fredj Tekaia et Michele Bidel,
352 pages, Réf. 0115

ALGORITHMES *par P. Beaufils, et W. Luther,*
296 pages, Réf. 0149

POUR UN CATALOGUE COMPLET DE NOS PUBLICATIONS

FRANCE
6-8, Impasse du Curé
75881 PARIS CEDEX 18
Tél. : (1) 42.03.95.95
Télex : 211801

U.S.A.
2344 Sixth Street
Berkeley, CA 94710
Tel. : (415) 848.8233
Telex : 336311

ALLEMAGNE
Vogelsanger. WEG 111
4000 Düsseldorf 30
Postfach N° 30.09.61
Tel. : (0211) 61 80 2-0
Telex : 08588163

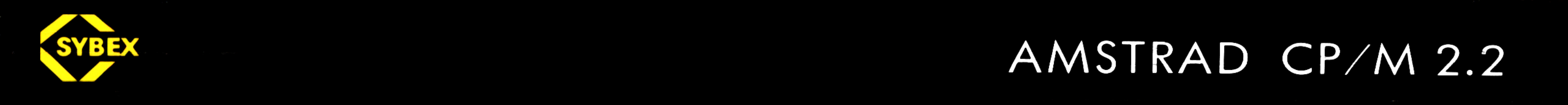


Paris • Berkeley • Düsseldorf

Achevé d'imprimer le 17 février 1986 sur les presses de l'Imprimerie «La Source d'Or»
63200 Marsat - Dépôt légal : 1er trimestre 1986 - Imprimeur n° 1974

CP/M est le système d'exploitation le plus répandu pour les micro-ordinateurs professionnels. Avec l'Amstrad, ce système est disponible pour la première fois sur un ordinateur coûtant moins de cinq mille francs, ouvrant à celui-ci l'immense bibliothèque des programmes d'applications professionnels. Ce livre décrit la version 2.2 de CP/M disponible sur les modèles CPC 464, CPC 664 et CPC 6128. Tous les aspects de CP/M 2.2 y sont étudiés, y compris l'utilisation des outils de développement livrés avec le système : l'éditeur ED, l'assembleur ASM et le programme de mise au point DDT. Un chapitre est consacré au programme de gestion des périphériques PIP. Toutes les commandes sont présentées par ordre alphabétique et décrites en détail avec la liste de leurs paramètres et leurs conditions d'utilisation. Douze annexes offrent un moyen de référence rapide aux utilisateurs déjà familiarisés avec CP/M.




SYBEX
AMSTRAD CP/M 2.2



Document numérisé avec amour par

AMSTRAD CPC

MÉMOIRE ÉCRITE



https://acpc.me/